

```

LIST L
;
; RAM ALLOCATION AND EQUATES
;
0000 = 0000 ORG 0
;
; DATA THAT DOESN'T REQUIRE INITIALIZATION.
;
0000 = 0020 DS 32 ; RESERVE 32 BYTES SCRATCH PAD AREA
0020 = 0040 PATHPTR DS 64 ; PATH/DIRECTION POINTER TABLE
0060 = 0001 JSDATA DS 1 ; JOYSTICK DATA
0061 = 0001 SCOREC DS 1 ; SCORE COUNT (USED TO INCREMENT SCORE BY SCRPD)
0062 = 0008 SNDPTR DS 8 ; POINTER TO SOUND DATA TABLE
006A = 0001 BDEADC DS 1 ; BUCK DEAD COUNTER
006B = 0002 ADDR DS 2 ; 2 BYTE POINTER TO SHAPE
006D = 0002 ADDR2 DS 2 ; 2 BYTE POINTER TO MASK
006E = 0001 BIT DS 1 ; X AXIS BIT POSITION WITHIN
0070 = 0001 BYTE DS 1 ; THE BYTE
0071 = 0001 HIGHT DS 1 ; JUST WHAT IT SAYS
0072 = 0002 MASK DS 2 ; POINTER FOR MASK USE
0074 = 0002 SCREEN DS 2 ; POINTER FOR SCREEN USE
0076 = 0002 SHAPE DS 2 ; POINTER FOR SHAPE USE
0078 = 0001 TIMER DS 1 ; 1 BYTE TIMER
0079 = 0001 YCOORD DS 1
007A = 0001 KBFLG DS 1
007B = 0001 LVLFLG DS 1
007C = 0001 MEANEG DS 1
007D = 0001 SPEEDC DS 1
007E = 0001 WIDTHC DS 1
007F = 0001 WIDTH DS 1
;
; DATA THAT NEEDS TO BE INITIALIZED TO 0 BEFORE EACH PLAY
;
0080 = 0001 LASTBS DS 1 ; LAST BUCK USED
0081 = 0001 BSECNT DS 1 ; COUNTER FOR BUCK FIRE REPEAT
0082 = 0001 POLCNT DS 1 ; POLE REPEAT COUNTER
0083 = 0001 SAUCNT DS 1 ; SAUCER DELAY COUNT (TIME REMAINING)
0084 = 0001 HOPCNT DS 1 ; HOPPER DELAY COUNT (TIME REMAINING)
0085 = 0004 SNDAGE DS 4 ; SOUND AGE
0089 = 0004 SDSTAT DS 4 ; SOUND STATUS
008D = 0001 POLEND DS 1 ; POLE FOUND FLAG/POLE X
008E = 0001 SSCORE DS 1 ; USED FOR CALCULATING ADDITIONAL SHIPS
008F = 0001 SCRATCH DS 1 ; USED FOR FUEL FLASH
0090 = 0001 HFRCNT DS 1 ; HOPPER FIRE COUNTS
0091 = 0001 PERCNT DS 1 ; POLE FIRE COUNTS
0092 = 0001 MERCNT DS 1 ; MOTHER ZORBA FIRE COUNT
0093 = 0001 STRCNT DS 1 ; STAR COUNT
;
; DATA TO BE INITIALIZED BEFORE EACH PLAY
;
0094 = 0001 SCRSET DS 1 ; SCROLL SET COUNTER
0095 = 0001 SCRCNT DS 1 ; SCROLL COUNT
0096 = 0001 MTNSPD DS 1 ; MOUNTAIN SPEED
0097 = 0001 MISRT DS 1 ; START OF MOUNTAIN DATA CARD #'S
0098 = 0001 MTNCNT DS 1 ; CURRENT COUNT OF MOUNTAIN
0099 = 0001 SCRSPD DS 1 ; SCROLL SPEED
009A = 0001 BUCKX DS 1 ; BUCK SHIP X POSITION
009B = 0001 BUCKY DS 1 ; BUCK SHIP Y POSITION
009C = 0003 FULAMT DS 3 ; AMOUNT OF FUEL REMAINING

```

*Testing circa 1983  
Donated by  
Charlie Kulor  
5-15-18*

; DATA TO BE INITIALIZED PRIOR TO EVERY NEW LEVEL OF PLAY

; BEGIN LEVEL VARIABLE DATA			
002F	= 0001	RSPONS	DS 1 ; JOYSTICK RESPONSE
00A0	= 0001	POLDLY	DS 1 ; DELAY BETWEEN POLES APPEARING
00A1	= 0001	SAIDLY	DS 1 ; TIME BETWEEN SAUCER SETS
00A2	= 0001	SAUSPD	DS 1 ; SAUCER SPEED
00A3	= 0001	HOPDLY	DS 1 ; TIME BETWEEN HOPPER SETS
00A4	= 0001	HOPSPD	DS 1 ; HOPPER SPEED
00A5	= 0001	MTRSPD	DS 1 ; MOTHER ZORBA SPEED
00A6	= 0001	MFQCNT	DS 1 ; NUMBER OF MFQ'S (STILL) TO KILL
00A7	= 0001	MINSPD	DS 1 ; MINIMUM SCROLL SPEED
00A8	= 0001	SPREAD	DS 1 ; POLE SPREAD
00A9	= 0001	HFRDLY	DS 1 ; HOPPER FIRE DELAY
00AA	= 0001	PRDLY	DS 1 ; POLE FIRE DELAY
00AB	= 0001	MFRDLY	DS 1 ; MOTHER ZORBA FIRE DELAY
00AC	= 0001	SYSTAT	DS 1 ; STATUS BITS FOR ROUND DESCRIPTION
	= 000E	PTCNT	EQU *-1VLDAT ; COUNT OF NUMBER OF ENTRIES

; DATA THAT NEEDS TO BE INITIALIZED BEFORE EACH GAME

00AD	= 0001	SHPLFT	DS 1 ; NUMBER OF SHIPS LEFT
00AE	= 0001	PLYLVL	DS 1 ; CURRENT PLAY LEVEL

; DATA THAT NEEDS TO BE INITIALIZED AFTER POWERUP ONLY

00AF	= 0001	FIRDLY	DS 1 ; DELAY FOR RAPID FIRE
00B0	= 0001	FIRFDG	DS 1 ; FUDGE AMOUNT USED FOR BUCK FIRE
00B1	= 003B	DLILOC	DS \$3B ; DLI ROUTINE!!!! YA-HOO!!

; VARIABLES IN STACK PAGE ARE CONTROLLED. THEY ARE NEVER  
 ; CHANGED BY ANY INITIALIZATION AND THEREFORE ARE HANDLED  
 ; BY THE EXECUTION OF THE PROGRAM. CSCORE AND HISCOR ARE  
 ; DONE THIS WAY SINCE REGULAR INITIALIZATION WOULD OTHERWISE  
 ; CLEAR THEM AND THERE VALUES ARE NEEDED FOR THE SIGN-ON  
 ; MESSAGE.

ASSERT \*C\$100

00EC	= 0900	ORG	\$900	
0900	= 0001	SDLSTL	DS 1	
0901	= 0001	SDLSTH	DS 1	
0902	= 0001	SDMCTL	DS 1	
0903	= 0001	CHBAS	DS 1	
0904	= 0001	CHART	DS 1	
0905	= 0001	SPRTA	DS 1	
0906	= 0001	SPRTB	DS 1	
0907	= 0002	COLDEG	DS 2	
0909	= 0004	HISCOR	DS 4 ; HI SCORE	
090B	= 0004	CSORE	DS 4 ; CURRENT SCORE	
0911	= 0001	COLOR0	DS 1	
0912	= 0001	COLOR1	DS 1	
0913	= 0001	COLOR2	DS 1	
0914	= 0001	COLOR3	DS 1	
0915	= 0001	COLOR4	DS 1	
0916	= 0001	PDLO	DS 1	
0917	= 0001	PDLI	DS 1	



ATARI HARDWARE EQUATES

```

    = E80A    RANDOM    EQU    $E80A
    = D40A    WSYNC     EQU    $D40A
    = D401    CHACTL    EQU    $D401
    = D402    CHBASE    EQU    $D402
    = D403    DLISTL    EQU    $D403
    = D400    DMACTL    EQU    $D400
    = E808    AUDCTL    EQU    $E808
    = C01F    CONSOLE   EQU    $C01F
    = E800    AUDE1     EQU    $E800
    = E801    AUDE1     EQU    $E801
    = E802    AUDE2     EQU    $E802
    = E803    AUDE2     EQU    $E803
    = E804    AUDE3     EQU    $E804
    = E805    AUDE3     EQU    $E805
    = E806    AUDE4     EQU    $E806
    = E807    AUDE4     EQU    $E807
    = C01D    GRCTL     EQU    $C01D
    = D40E    NMEN      EQU    $D40E
    = D302    PACTL     EQU    $D302
    = D303    PBCTL     EQU    $D303
    = D407    PMBASE    EQU    $D407
    = 0206    VDSLST    EQU    $0206
    = 0202    VVBLKI    EQU    $0202
    = 0800    RTNADD     EQU    $800
    = 1000    RELSCR     EQU    $1000
    = E80F    SKCTL     EQU    $E80F
    = D40B    VCOUNT     EQU    $D40B
    = E80E    SKSTAT     EQU    $E80E
    = E809    KBCODE     EQU    $E809
    
```

: RETURN ADDRESS FOR GOSUB IN PATH ROUTINE  
 : RELOCATED SCROLL COLOR ADDR

RANDOM OTHER EQUATES

```

    = 0600    OBJTBL     EQU    $600
    
```

```

0918 = 2FFD    ORG    $2FFD
2FFD FF        DB    $FF
2FFF 0060      DW    START
    
```

SYSTEM STARTUP

```

A000 = 6000    ORG    $6000
6000          START
    
```

CLEAR HIGH SCORE AND SCORE

```

6000 AD0709    LDA    COLDFFG
6003 C9A5      CMP    #$A5
6005 D00A ^6011 BNE    COLDST
6007 AD0809    LDA    COLDFFG+1
    
```

```

600A C95A      CMP      #$5A
600C D003 ^6011  BNE     COLDST
600E 4CE568     JMP     CINIT
6011          COLDST
6011 A200      LDA     #0
6013 A203      LDX     #3
6015          CLERHI
6015 9D0909     STA     HISCDR,X
6018 9D0D09     STA     CSCORE,X
601B BA        DEY
601C 10F7 ^6015  BPL     CLERHI
601E A9A5      LDA     #$A5
6020 8D0709     STA     COLDFG
6023 A95A      LDA     #$5A
6025 8D0809     STA     COLDFG+1
6028 4CE568     JMP     CINIT      ; DO PWR UP INIT
602B          DRIVER
602B 203474     JSR     JOYSTK      ; READ JOYSTICK
602E 204A60     JSR     CONVSP     ; MAKE MINSPD EFFECT SCROLL SPEED
6031 203D6B     JSR     MTNWRK     ; MOVE MOUNTAINS
6034 206163     JSR     SOUND      ; SOUNDS
6037 209C6F     JSR     BSETRE     ; LET FIRE FASTER
603A 207E6E     JSR     MOVE       ; MOVE OBJECTS ON DISPLAY
603D 206163     JSR     SOUND      ; SOUNDS AGAIN
6040 20236C     JSR     BSWORK     ; BS PROCESSING
6043 20ED68     JSR     POLES      ; DO POLE PROCESSING
6046 20CD6B     JSR     SAUCER     ; DO SAUCER PROCESSING
6049 207E62     JSR     HOPPER     ; DO HOPPER PROCESSING
604C 202664     JSR     COLLISN    ; DO COLLISION DETECTION
604E 20DB63     JSR     SCORE      ; DISPLAY SCORE
        ; JSR     UFODSP          ; ***???***
6052 20C962     JSR     FUEL       ; DO FUEL PROCESSING
6055 209A61     JSR     HOPFIR     ; HOPPER FIRE ROUTINE
6058 20F961     JSR     POLEFIR    ; POLE FIRE ROUTINE
605B 20FC60     JSR     MOTHER     ; PROCESS MOTHER ZORBA
605E 20B960     JSR     STARS      ; PROCESS STARS
6061 200000     JSR     PAUSE      ; CHECK FOR PROGRAM PAUSE
6064 209B63     JSR     BSOUND
6067 4C2B60     JMP     DRIVER
606A 48        CONVSP  PHA
606B 8A        TXA
606C 48        PHA
606D A599     LDA     SCRSPD      ; KLUDGE FOR ATARI CONVERSION
606F A208     LDX     #8
6071 DDE340     CVLDFG  CMP     SCNVF-1,X
6074 9007 ^607D  BCC     CONVOK
6076 F005 ^607D  BEQ     CONVOK
6078 CA        DEY
6079 D0F6 ^6071  BNE     CVLDFG
607B A201     LDX     #1
607D 8ECA00     CONVOK  STX     SPEED
6080 68        PLA
6081 AA        TAX
6082 68        PLA
6083 60        RTS

```

0028887868 SCNVF DB \$AD,\$98,\$88,\$78,\$68,\$58,\$48,\$38

ROUTINE DISPLAYS UFO'S REMAINING TO BLAST

Address	Label	Operation	Value
608C		UFOCNT	
608C	A5A6	LDA	UFOCNT
608E	F020 ^60B0	BEQ	UFOXT
6090	C913	CMF	#19
6092	901D ^60B1	BCC	UFEI
6094	A956	LDA	#22 OR 64
6096	8D6420	STA	\$2064
6099	A912	LDA	#18
609B	8500	STA	0
609D	A900	LDA	#0
609F	A213	LDX	#19
60A1	9D6420	STA	\$2064,X
60A4	CA	DEX	
60A5	F009 ^60B0	BEQ	UFOXT
60A7	E400	CPX	0
60A9	D0E6 ^60A1	BNE	UFOIX
60AB	A915	LDA	#21
60AD	ACA160	JMP	UFOIX
60B0	60	RTS	
60B1	A000	LDY	#0
60B3	8C6420	STY	\$2064
60B6	4C9B60	JMP	UFOOK

ROUTINE CREATES STARS AT CONSTANT TIME PERIODS, BUT  
 DEPENDS ENOUGH THAT THE DISPLAY CONSTANTLY HAS  
 LOTS OF STARS IN IT.

```

60B9 STARS
60B9 A5AC LDA SYSTAT ; SEE IF SPACE SCENE
60BB 2902 AND #2
60BD D001 ^60C0 BNE STR20 ; YES - CREATE STARS
60BE STR10
60BE 60 RTS
60C0 STR20
60C0 E693 INC STRCNT ; SEE IF TO CREATE ONE THIS TIME
60C2 A593 LDA STRCNT
60C4 4A LSR A
60C5 20E8 ^60BE BCC STR10 ; NO
60C7 A209 LDY #9 ; TRY TO CREATE A STAR
60C9 20E06C JSR CREATE
60CC D0F1 ^60BE BNE STR10 ; NO ROOM - GO
60CE STR30
60CE ADOAE8 LDA RANDOM ; X=64-121 (128 POSITIONS)
60D1 30E8 ^60CE BMI STR30
60D3 18 CLC
60D4 6940 ADC #64
60D6 990106 STA OBJTBL+1,Y
60D9 08 PHP ; SAVE SIGN OF X
60DA STR40
60DA ADOAE8 LDA RANDOM ; GET PATH BASE 1-5
60DD 2907 AND #7
60DE F0E9 ^60DA BEQ STR40
60E1 C906 CMP #6 ; ONLY 5
60E3 B0F5 ^60DA BCS STR40
60E5 18 CLC ; NORMALIZE
60E6 6913 ADC #19 ; ACTUAL PATH #15-20-30
60E8 28 PLP ; GET SIGN OF X
60E9 3003 ^60EE BMI STR50 ; NEGATIVE X'S GET 20-24
60EB 18 CLC
60EC 6906 ADC #6 ; POSITIVE X'S GET 25-29
60EE STR50
60EE 20E6AD JSR SETPTH ; SET PATH
60F1 A936 LDA #54 ; Y CENTERED (ALMOST)
60F3 990206 STA OBJTBL+2,Y
60F6 A930 LDA #30 ; SPEED
60F8 990506 STA OBJTBL+5,Y
60FB 60 RTS
    
```

ROUTINE BASICALLY JUST MAKES SURE THAT WHEN MOTHER  
ZORBA IS SUPPOSED TO BE OUT THERE, SHE IS.

```

60EC      MOTHER
60EC A5AC      LDA      SYSTAT      ; SEE IF SUPPOSED TO BE A ZORBA
60EE 2904      AND       #4
6100 D001 ^6108 BNE      MOTH20      ; YES
6102      MOTH10
6102 60        RTS
6103      MOTH20
6103 A000      LDY       #0          ; SEE IF MOTHER ZORBA OUT THERE
6105      MOTH30
6105 B90006     LDA      OBJTBL,Y
6108 C906      CMR       #6
610A E040 ^614C BEQ      MOTH70      ; GOT ONE - SEE IF TO FIRE
610C 98        TYA
610D 18        CLC
610E 6908      ADC      #8
6110 A8        TAY
6111 C000      CPY       #0
6113 D0F0 ^6105 BNE      MOTH30
;
; ZORBA'S SUPPOSED TO BE OUT THERE AND SHE AIN'T. TRY TO PUT HER THERE.
;
6115 A206      LDX      #6
6117 20E06C     JSR      CREATE
611A D0E6 ^6102 BNE      MOTH10      ; CAN'T - OH WELL
611C A5AC      LDA      SYSTAT      ; SEE IF SPACE/GROUND
611E 2902      AND      #2
6120 E01F ^6141 BEQ      MOTH60      ; GROUND PATHS DIFFERENT THAN SPACE PATHS
6122 AD0AE8     LDA      RANDOM      ; SELECT PATH 15/16
6125 3005 ^612C BMI      MOTH40      ; USE PATH 15
6127 A910      LDA      #16          ; USE PATH 16
6129 4C2E61     JMP      MOTH50
612C      MOTH40
612C A90F      LDA      #15
612E      MOTH50
612E 200E6D     JSR      SETPTH
6131 A980      LDA      #80          ; X,Y,SPEED
6133 990106     STA      OBJTBL+1,Y
6136 A990      LDA      #144
6138 990206     STA      OBJTBL+2,Y
613B A5A5      LDA      MTRSPD
613D 990506     STA      OBJTBL+5,Y
6140 60        RTS
6141      MOTH60
6141 AD0AE8     LDA      RANDOM      ; GROUND PATHS USE SAUCER PATHS FOR NOW
6144 2903      AND      #3
6146 18        CLC
6147 6908      ADC      #3
6149 4C2E61     JMP      MOTH50
;
; ZORBA OUT THERE - SEE IF TO FIRE
;
614C      MOTH70
614C B90706     LDA      OBJTBL+7,Y
614E 2910      AND      #10
6151 D001 ^6154 BNE      MOTH90      ; ATTACKING - MAYBE
6153      MOTH80

```



```

6153 60 RTS
6154 MOTH80
6154 A592 LDA MERCNT
6154 E003 ^A15B BEQ MOT100 : FIRE!
6158 C692 DEC MERCNT : DEC COUNT
615A 60 RTS
615B MOT100
615B 840B STY #1 : SAVE INDEX TO ZORBA
615D A5AB LDA MFRPLY : RESET COUNTER FOR NEXT TIME
615E 8592 STA MERCNT
6161 B90306 LDA OBJTBL+3,Y : IF ZORBA SIZE<2, NO FIRE ALLOWED
6164 C902 CMP #2
6166 90EB ^A153 BCC MOTH80 : FORGET IT - RAM BUCK INSTEAD?
6168 A204 LDX #4 : START ENEMY SHOT
616A 20E06C JSR CREATE
616D E0E4 ^A153 BNE MOTH80 : NO ROOM
616F A90C LDA #12 : USE PATH12 (LIKE HOPPER FIRE)
6171 200E6D JSR SETPTH
6174 A60B LDX #1 : SET UP TO GET ZORBA X+INDEX DATA
6176 BD0306 LDA OBJTBL+3,X : ZORBA SIZE USED FOR INDEX
6179 990306 STA OBJTBL+3,Y : SHOT SIZE = ZORBA SIZE
617C 6A TAX
617D BDA466 LDA MTRHTR,X : X OFFSET TO CENTER
6180 A60B LDX #1 : ADD TO ZORBA X
6182 18 CLC
6183 7D0106 ADC OBJTBL+1,X
6186 990106 STA OBJTBL+1,Y : SHOT X
6189 BD0206 LDA OBJTBL+2,X : Y=ZORBA Y
618C 990206 STA OBJTBL+2,Y
618F A220 LDA #20 : SPEED
6191 990506 STA OBJTBL+5,Y
6194 A903 LDA #3 : SOUND SAME AS HOPPER FIRE
6196 20A863 JSR SNDINI
6199 60 RTS
  
```



ROUTINE CHECKS SYSTEM STATUS TO SEE IF HOPPERS ARE  
TO FIRE. IF SO, THE HERDLY IS USED TO DETERMINE HOW  
FREQUENTLY THEY WILL FIRE.

```
619A      HOPEIR
619A A5AC      LDA      SYSTAT      ; SEE IF HOPPERS FIRE THIS ROUND
619C 1004 ^61A4 BPL      HOPE10     ; NO
619E A590      LDA      HFRONT     ; SEE IF TO FIRE THIS TIME
61A0 E003 ^61A5 BEQ      HOPE20     ; YES
61A2 C690      DEC      HERCNT
61A4      HOPE10
61A4 60        RTS
61A5      HOPE20
61A5 A010      LDY      #16        ; TRY TO FIND A HOPPER
61A7      HOPE30
61A7 B90006     LDA      OBJTBL,Y
61AA C903      CMP      #3
61AC E00C ^61BA BEQ      HOPE40     ; GOT ONE
61AE      HOPE35
61AE 98        TYA
61AF 18        CLC
61B0 6908      ADC      #8
61B2 A8        TAY
61B3 C980      CMP      #$80
61B5 D0F0 ^61A7 BNE      HOPE30
61B7 4CF461     JMP      HOPE50     ; NONE - EXIT
61BA      HOPE40
61BA B90106     LDA      OBJTBL+1,Y ; FIRE ONLY IF IN DISPLAY AREA
61BD C928      CMP      #40        ; X MUST BE 40-215
61BE 90ED ^61AE BCC      HOPE35
61C1 89D7      CMP      #215
61C3 80E9 ^61AE BCS      HOPE35
61C5 841E      STY      $1F        ; SAVE Y
61C7 A204      LDX      #4        ; 4 IS ENEMY SHOT
61C9 20E06C     JSR      CREATE     ; TRY TO CREATE A SHOT
61CC D0D6 ^61A4 BNE      HOPE10     ; NO ROOM - FORGET IT
61CE A90C      LDA      #12        ; THESE ARE ON PATH12
61D0 200E6D     JSR      SETPTH
61D3 A81E      LDX      $1F        ; GET INDEX TO SOURCE OF SHOT
61D5 BD0106     LDA      OBJTBL+1,X ; GET X+4 FOR START
61D8 18        CLC
61D9 6904      ADC      #4
61DB 990106     STA      OBJTBL+1,Y
61DE BD0206     LDA      OBJTBL+2,X ; Y
61E1 990206     STA      OBJTBL+2,Y
61E4 BD0306     LDA      OBJTBL+3,X ; SIZE=HOPPER SIZE
61E7 990306     STA      OBJTBL+3,Y
61EA A980      LDA      #$80       ; SPEED
61EC 990506     STA      OBJTBL+5,Y
61EF A903      LDA      #3        ; ENEMY FIRE SOUND
61F1 20A863     JSR      SNDINT
61F4      HOPE50
61F4 A5A9      LDA      HERDLY
61F6 8590      STA      HERCNT
61F8 60        RTS
```

POLE FIRE ROUTINE CHECK TO SEE IF THEY SHOULD FIRE SHOTS

```

61F9  ENIFIR
61F9  A5AC      LDA      SYSTAT
61FB  2960      AND      #$60
61FD  F004 ^6205 BEQ      POLE10      ; NOT TO FIRE
61FF  A591      LDA      PFRCNT
6201  F003 ^6206 BEQ      POLF20      ; FIRE THIS TIME!
6203  E691      DEC      PERCNT
6205  POLF10
6205  60        RTS
6206  POLF20
6206  A010      LDY      #16      ; LOOK FOR A POLE
6208  POLF30
6208  B90006     LDA      OBJTBL,Y
620B  C901      CMP      #1
620D  F00E ^621D BEQ      POLE50      ; GOT ONE
620F  POLF40
620F  58        TYA
6210  18        CLD
6211  6208      ADD      #8
6213  A8        TAY
6214  C080      CPY      #$80
6216  D0F0 ^6208 BNE      POLF30
6218  POLF45
6218  A5AA      LDA      PFRDLY      ; RESET DELAY TO NEXT FIRE
621A  8591      STA      PERCNT
621C  60        RTS
621D  POLF50
621D  841F      STY      $1F
621F  B90306     LDA      OBJTBL+3,Y      ; ONLY FIRE ON SIZES 0-6
6222  E907      CMP      #7
6224  D0E9 ^620F BCS      POLF40      ; NO - FORGET THIS ONE
6226  A10AE8     LDA      RANDOM
6229  4A        LSR      A      ; SEE IF TO FIRE IN/OUT
622A  9042 ^626E BCC      POL110      ; FIRE IN
622C  A5AC      LDA      SYSTAT      ; SEE IF OK TO FIRE OUT
622E  2920      AND      #$20
6230  F0DD ^620F BEQ      POLF40      ; NO - TRY ANOTHER ONE
6232  B90706     LDA      OBJTBL+7,Y      ; SEE IF THIS A LEFT OR RIGHT POLE
6235  2908      AND      #8
6237  D005 ^623E BNE      POLF70      ; LEFT POLE
6239  POLF60
6239  A90E      LDA      #14      ; CREATE RIGHT SHOT
623B  4C4062     JMP      POLF80
623E  POLF70
623E  A90D      LDA      #13      ; CREATE LEFT SHOT
6240  POLF80
6240  851E      STA      $1E      ; SAVE PATH
6242  A208      LDY      #8      ; TRY TO CREATE POLE SHOT
6244  20E06C     JSR      CREATE
6247  D0CE ^6218 BNE      POLF45      ; NO ROOM - FORGET THIS RIGHT NOW
6249  A51E      LDA      $1E      ; SET PATH
624B  200E6D     JSR      SETPTH
624E  A61F      LDY      $1F      ; INDEX INTO SOURCE LOCATION
6250  BD0106     LDA      OBJTBL+1,X      ; SET X
6253  970106     STA      OBJTBL+1,Y
6256  POLF85
6256  BD0206     LDA      OBJTBL+2,X      ; SET Y=BASE ADDRESS+16

```

## POLE FIRE ROUTINE

J:5200 .A55

6259	18	CLC		
625A	6910	ADC	#16	
625E	990206	STA	OBJTBL+2,Y	
625F	A980	LDA	#80	: SPEED
6261	990506	STA	OBJTBL+5,Y	
6264	A904	LDA	#4	: RAY SOUND
6266	20A863	JSR	SNDINI	
6269				
6269	A41F	LDY	\$1F	: RESTORE Y
626B	4C0E62	JMP	POLE40	: NEXT
626E				
626E	A5AC	LDA	SYSTAT	: SEE IF OK TO FIRE IN
6270	2940	AND	#840	
6272	E09B ^620F	BEQ	POLE40	: NO - TRY ANOTHER
6274	B90706	LDA	OBJTBL+7,Y	: SEE IF LEFT/RIGHT POLE
6277	2908	AND	#8	
6279	D0BE ^6239	BNE	POLE60	: LEFT - FIRE RIGHT
627B	4C3E62	JMP	POLE70	: RIGHT - FIRE LEFT

ROUTINE TO PROCESS (START) HOPPERS AT PERIODIC INTERVALS ON  
VARYING PATHS

627F HOPPER

627E A5AC LDA SYSTAT ; SEE IF THEY EXIST  
6280 2910 AND #10  
6282 F006 ^628A BEQ HOPP10 ; NO  
6284 A584 LDA HOPCNT ; SEE IF OK TO START ONE  
6286 E003 ^628B BEQ HOPP20 ; YEP  
6288 C684 DEC HOPCNT  
628A HOPP10  
628B 60 RTS

628B HOPP20

628B A203 LDX #3 ; TRY TO START A HOPPER  
628D 20E06C JSR CREATE  
6290 00F8 ^628A BNE HOPP10 ; NO ROOM

6292 HOPP30

6292 A00AE8 LDA RANDOM ; GET ONE OF TEN PATHS  
6295 2901 AND #1 ; PATH 10-11  
6297 18 CLC  
6298 690A ADC #10 ; NORMALIZE FOR PATHS (C=0)  
629A 200E6D JSR SETPTH ; SET PATH IN TABLE

629D HOPP40

629D A908 LDA #8 ; SET SIZE  
629F 990306 STA QB,ITBL+3,Y  
62A2 A00AE8 LDA RANDOM ; SET RANDOM X  
62A5 990106 STA QB,ITBL+1,Y

62A8 HOPP50

62A8 A5A4 LDA HOPSPD ; SPEED  
62AA 990506 STA QB,ITBL+5,Y  
62AD A5A3 LDA HOPDLY ; RESET COUNTER  
62AF 8584 STA HOPCNT  
62B1 60 RTS ; DONE

ROUTINE DISPLAYS NUMBER OF SHIPS REMAINING. UPDATES ACTUAL  
EVERY TIME SINCE THIS IS SO FAST (CHARACTER GRAPHICS MODE).

62B2	SHIPS		
62B2	A5AD	LDA	SHPLET
62B4	38	SEC	
62B5	E901	SBC	#1
62B7	F00F ^62C8	RED	JAMIT
62B9	C906	CMP	#6
62BB	9002 ^62BF	BCC	SHOK
62BD	A905	LDA	#5
62BF	AA	TAX	
62C0	A914	LDA	#20
62E2	9D3520	STX	\$2035,X
62C5	CA	DEX	
62C6	D0FA ^62C2	BNE	SHOPD
62C8	60	RTS	



ROUTINE DECREMENTS FUEL AND UPDATES DISPLAY  
GRAPH WHENEVER IT CHANGES.  
WAS TIME GRAPH BUT GRAPH NOW REPRESENTS FUEL WHICH DECREMENT  
AT A RATE INVERSLY TO THE SCROLL SPEED.

62C9

FUEL

NOW DO FUEL GONE CHECK AND FUEL DECREMENT

62C9

FUEL03

62C9 A59E

LDA

FUELAMT+2 ; SEE IF ALREADY OUT OF FUEL

62CB 059D

ORA

FUELAMT+1

62CD 059C

ORA

FUELAMT

62CF 0001 ^62D2

BNE

FUEL07

; NOPE

62D1

FUEL05

62D1 60

RTS

62D2

FUEL07

62D2 A599

LDA

SCRSPD

; DECREMENT FUEL BY FACTOR OF -SCRSPD

62D4 49FE

EOR

#0FF

62D6 18

CLC

62D7 6901

ADC

#1

62D9 A214

LDX

#20

62DB 20C273

JSR

MULT

62DE 8600

STX

0

62E0 8501

STA

1

62E2 A59C

LDA

FUELAMT

62E4 38

SEC

62E5 E500

SRC

0

62E7 859C

STA

FUELAMT

62E9 A59D

LDA

FUELAMT+1

62EB E501

SRC

1

62ED 859D

STA

FUELAMT+1

62EF B0E0 ^62D1

BCS

FUEL05

; NO CHANGE IN MSB - JUST EXIT

62F1 A59E

LDA

FUELAMT+2

62F3 E900

SRC

#0

62F5 859E

STA

FUELAMT+2

62F7 B008 ^6301

BCS

FUEL10

62F9 A900

LDA

#0

; FORCE FUEL=0

62FB 859C

STA

FUELAMT

62FD 859D

STA

FUELAMT+1

62FF 859E

STA

FUELAMT+2

6301

FUEL10

---> DISPLAY FUEL GRAPH

FUELAMT DETERMINES HOW MUCH FUEL PLAYER HAS. USE MSB (FUEL+2)

FOR THE GRAPH WHICH HAS A RANGE OF 00-\$77.

6301 A59E

LDA

FUELAMT+2

6303 4A

LSR

A

6304 4A

LSR

A

6305 4A

LSR

A

6306 C90D

CMP

#13

6308 9002 ^630C

BCS

FUEL0K

630A A90C

LDA

#12

630C AA

FUEL0K

TAX

630E A8

TAX

630E F008 ^6318

BEQ

RFHG

6310 A9E0

LDA

#32 OR \$C0



```

6312 9D4220      FULP      STA      $2028+20+6,X
6315 6A          DEX
6316 00FA ^6312   BNE      FULP
6318 18          REHG      INY
6319 98          TYA
631A AA          TAX
631B A59E        ZODS      LDA      FULAMT+2
631D 2907        AND      #%00000111
631F F007 ^6328   BEQ      FUXX
6321 18          CLC
6322 62D8        ADC      #24 DR $C0
6324 9D4220      STA      $2028+20+6,X
6327 E8          ZOD      INX
6328 E000        FUXX      CPX      #13
632A R009 ^6335   BCS      FUXD
632C A2D8        LDA      #24 DR $C0
632E 9D4220      STA      $2028+20+6,X
6331 E8          INX
6332 4C2863      JMP      FUXX
        :
        : ----- SET COLOR OF GRAPH
        :
        : USE FULAMT+2 AGAIN. $00-$18=RED      $19-$30=YELLOW
        :
        : $31-$77=GREEN (>31)
        :
6335 A59E        FUXD      LDA      FULAMT+2
6337 0915        CMP      #15
6339 B017 ^6352   BCS      TYEL
633B A906        LDA      #6
633D 20A863      JSR      SNDINI
6340 A58F        LDA      SCRATCH
6342 4901        EOR      #1
6344 858F        STA      SCRATCH
6346 D005 ^634D   BNE      BLNKIT
6348 A90F        LDA      #0F
634A 4C5D63      JMP      BYEBYE
634D A934        BLNKIT   LDA      #34
634F 4C5D63      JMP      BYEBYE
6352 C930        TYEL     CMP      #30
6354 R005 ^635B   BCS      TGRN
6356 A91E        LDA      #1E
6358 4C5D63      JMP      BYEBYE
635B A9E6        TGRN     LDA      #E6
635D 8D1409      BYEBYE   STA      COLOR3
6360 60          RTS

```

SOUND DRIVER

ROUTINE LOOKS TO SEE IF SOUND GENERATOR IN USE. IF IT IS,  
IT STEPS THROUGH THE FREQUENCIES.

SOUND

```

6361 A206 LDX #6 ; INC DATA POINTER INDEX
6363 A003 LDY #3 ; INC COUNT/RATE TABLES INDEX
6365 SOUN10 LDA SDSTAT,Y ; SEE IF THIS CHANNEL ACTIVE
6365 B98900 BEQ SOUN50 ; NO
6368 E02B ^6395 SOUN20 JSR INSDPT ; GET NEXT DATA BYTE
636A 208273 LDA [SNDPTR,X1] ; KILL IT
636D A162 BEQ SOUN40 ; SEE IF CONTROL BYTE
636F F01E ^638F CMP #1 ; YEP
6371 C901 BEQ SOUN30 ; SET FREQUENCY
6373 E00E ^6384 STA AUDE1,X ; BIRTHDAY TIME
6375 9D00E8 LDA SNDAGE,Y
6378 B98500 CLC
637B 18 ADC #1
637C 6901 STA SNDAGE,Y
637E 998500 JMP SOUN50
6381 4C9563 SOUN30 JSR INSDPT ; GET CONTROL BYTE
6384 208273 LDA [SNDPTR,X1]
6387 A162 STA AUDE1,X ; NOW SET FREQ
6389 9D01E8 JMP SOUN20
638E SOUN40 STA SDSTAT,Y ; KILL THIS AND FREE CHANNEL
638F 998900 STA AUDE1,X
6392 9D01E8 SOUN50 DEX
6395 CA DEX
6396 CA DEY
6397 85 BPL SOUN10
6398 10CR ^6365 RTS
639A 60

```

```

1 639B BOUND
2 639B A922 LDA #22
3 639D 8001E8 STA ADR1
4 63A0 A599 LDA SCRSPD
5 63A2 49EF EOR #FF
6 63A4 8D00E8 STA ADR1
7 63A7 60 RTS
8
9
10 ; ROUTINE LOOKS FOR UNUSED SOUND CHANNEL. IF IT FINDS
11 ; ONE, THAT CHANNEL IS USED. IF NOT, A SEARCH IS MADE FOR THE
12 ; OLDEST CHANNEL. THE SOUND TO USE IS GIVEN IN A
13
14 63A8 SNDINI
15 63A8 48 PHA ; SAVE SOUND # TO USE
16 63A9 A203 LDX #3 ; LOOK FOR ANY OPEN CHANNEL
17 63AB SNDI10
18 63AB B589 LDA SDSTAT,X
19 63AD D005 ^63B4 BNE SNDI20
20 63AF 8600 STX 0 ; FOUND ONE - FORCE THIS AS ONE TO USE
21 63B1 4CBE63 JMP SNDI50
22 63B4 SNDI20
23 63B4 CA DEX
24 63B5 D0F4 ^63AB BNE SNDI10 ; KEEP LOOKING
25 63B7 68 PLA
26 63B8 48 PHA
27 63B9 AA TAX
28 63BA BDD174 LDA PRECHN,X ; PREFERED CHANNEL
29 63BD 8500 STA 0
30 63BE SNDI50
31 63BF A600 LDX 0 ; SET STAT BYTE FOR CHANNEL IN USE
32 63C1 A9FF LDA #FF
33 63C3 9589 STA SDSTAT,X
34 63C5 A900 LDA #0 ; RESET AGE
35 63C7 9585 STA SNDAGE,X
36 63C9 0600 ASI 0 ; PREPARE FOR SNDPTR INDEX
37 63CB A600 LDX 0
38 63CD 68 PLA ; GET SOUND TO USE
39 63CE 0A ASI A ; DOUBLE FOR TABLE INDEX
40 63CF A8 TAY
41 63D0 B9B074 LDA SNDIBL,Y
42 63D3 9562 STA SNDPTR,X
43 63D5 B9B174 LDA SNDIBL+1,Y
44 63D8 9563 STA SNDPTR+1,X
45 63DA 60 RTS ; DONE
46
47 ; SCORE IS INCREMENTED EACH PASS BY SCRSPD-37.64THS
48 ; OF A POINT EACH PASS. =(OLD COUNT+SCRSPD-37)/64-(OLD COUNT/64)
49
50 ; NEW SHIP IS AWARDED EVERY 20000 PTS
51
52 63DB SCORE
53 63DB A56A LDA BDEADC
54 63DD D022 ^6401 BNE SCOR01 ; DON'T DO THIS WHEN BUCK IS DEAD
55 63DF A561 LDA SCOREC ; GET OLD COUNT
56 63E1 4A LSR A ; /64
57 63E2 4A LSR A
58 63E3 4A LSR A
59 63E4 4A LSR A

```

SOUND INITIALIZATION ROUTINE

```

63E5 4A      LSR      A
63E6 4A      LSR      A
63E7 8501    STA      1      ; SAVE IT
63E9 A529    LDA      SCRSPD  ; GET SCRSPD
63FB 38      SEC
63EC E237    SBC      #37    ; NORMALIZE FOR MIN VALUE
63FE 18      CLC
63FF 6561    ADC      SCOREC  ; ADD IN COUNTS
63F1 8561    STA      SCOREC  ; UPDATE COUNTS
63F3 6A      ROR      A      ; KEEP CARRY AND /64
63F4 4A      LSR      A
63F5 4A      LSR      A
63F6 4A      LSR      A
63F7 4A      LSR      A
63F8 4A      LSR      A
63F9 38      SEC
63FA E501    SBC      1
63FC A200    LDY      #0      ; X.A IS INCREMENT VALUE FOR SCORE
63FE 208273  JSR      ADDSCR  ; ADD TO SCORE
; MAKE CHECK FOR NEW SHIP AWARD
6401          SCOR01
6401 AD0E09  LDA      CSCORE+1 ; GET CHANGE IN DIGIT 2
6404 F8      SED
6405 38      SEC
6406 E58E    SBC      SSCORE
6408 08      CLD
6409 C902    CMP      #2
640B 9015 ^6422 BCC      SCOR05      ; NOT ENOUGH
640D E6AD    INC      SHPLFT  ; GIVE ANOTHER SIP
640F D002 ^6413 BNE      SCOR02
6411 C6AD    DEC      SHPLFT  ; MAX AT FF
6413          SCOR02
6413 AD0E09  LDA      CSCORE+1 ; SET SSCORE FOR NEXT PASS
6416 29FE    AND      #3FE
6418 858E    STA      SSCORE
641A A90A    LDA      #3A
641C 20A863  JSR      SNDINI
641E 20B262  JSR      SHIPS      ; DISPLAY CHANGE
6422          SCOR05
6422 203F71  JSR      SCRUPD  ; DISPLAY SCORE
6425 60      RTS

```

COLLISION DETECT ROUTINE

```

; ROUTINE SCANS OBJECT DESCRIPTORS FOR BUCK SHOTS. IF ONE
; IS FOUND, A CHECK IS MADE FOR COLLISIONS WITH ANY DROIDS,
; SAUCERS, OR MOTHER ZORBA HERSELF.
; ENTER:
; EXIT:
; REGISTER USAGE:

```

```

; $OB = OBJECT LENGTH
; $OC = OBJECT WIDTH
; $OD = DESTROYED

```

```

; $10 = LENGTH
; $11 = WIDTH
; $12 = BUCK SHOT X
; $13 = BUCK SHOT Y
; $1E = OBJECT INDEX SAVE
; $1F = BUCK SHOT INDEX SAVE

```

```

1 6426 COL1SN
2
3 6426 A000 LDY #0 ; SEE IF ANY BUCK SHOTS
4
5 6428 COL110
6 6428 B90006 LDA OBJTBL,Y
7 6428 C905 CMP #5
8 642D F00D ^643C BEQ COL130 ; GOT ONE
9 642E COL120
10 642E 98 TYA ; GO TO NEXT ONE
11 6430 18 CLC
12 6431 6908 ADC #8
13 6433 A8 TAY
14 6434 C900 CMP #500
15 643A D0E0 ^6428 BNE COL110
16 6438 4C3065 JMP COL120 ; GO CHECK IF BUCK DEAD DUCK
17 643B COL125
18 643B 60 RTS ; DONE
19
20 ; GOT BUCK SHOT - SET UP PARAMETERS
21
22 643C COL130
23 643C 841F STY $1E ; SAVE CURRENT INDEX
24 643E B90106 LDA OBJTBL+1,Y ; GET X
25
26 6441 8512 STA $12
27 6443 B90206 LDA OBJTBL+2,Y ; GET Y
28 6446 38 SEC ; SUBTRACT OUT Y POSITION FUDGE AMOUNT
29 6447 E5B0 SBC FIREDG
30 6449 8513 STA $13
31 644B A20A LDY #10 ; X=TYPE*2 FOR INDEX
32 644D 20506E JSR ROscal ; GET LENGTH/WIDTH
33 6450 A50B LDA $B ; LENGTH+LENGTH FUDGE TO 10
34 6452 18 CLC
35 6453 65B0 ADC FIREDG
36 6455 8510 STA $10
37 6457 A50C LDA $C ; WIDTH TO 11
38 6459 8511 STA $11

```

```

; SEARCH FOR SAUCERS/DROIDS(HOPPERS)/MOTHER ZORBA

```



645B	A000	LDY	#0	: INIT INDEX
645D	COL140			
645D	B90006	LDA	OBJTBL,Y	: SEE IF DROID/SAUCER
6460	C902	CMP	#2	
6462	E01E ^6482	BEQ	COL170	: SAUCER
6464	C903	CMP	#3	
6466	E01A ^6482	BEQ	COL170	: DROID
6468	C906	CMP	#6	
646A	F00E ^647A	BEQ	COL165	: MOTHER ZORBA
646C	COL150			
646C	98	TYA		: GO TO NEXT
646D	18	CLC		
646E	A908	ADC	#8	
6470	A8	TAY		
6471	C000	CPY	#000	
6473	D0E8 ^645D	BNE	COL140	
6475	COL160			
6475	A41F	LDY	\$1F	: NO MORE - RESTORE FOR NEXT BUCK SHOT
6477	4C2F64	JMP	COL120	
647A	COL165			
647A	20AD66	JSR	CKCRSH	: SEE IF COLLISION WITH ZORBA
647D	F0ED ^646C	BEQ	COL150	: NO
647E	4C0066	JMP	COL190	: MAYBE
: GOT SAUCER, DROID, OR ZORBA. GET DISTANCES BETWEEN X,Y COORDINATES AND				
: COMPARE THAT WITH THE LENGTH/WIDTH OF THE OBJECTS				
6482	COL170			
6482	20AD66	JSR	CKCRSH	
6485	F0E5 ^646C	BEQ	COL150	: NO COLLISION
: COLLISION DETECTED - CANCEL OBJECT BY FORCING PATH AS EXPLOSION				
: PATH (PATH7). CANCEL BUCK SHOT BY FORCING PATH TO CANCEL PATH				
: (PATH8). IN BOTH CASES, RPTCNT SET TO 0 TO FORCE PATH EXECUTION				
: IMMEDIATELY.				
6487	COL110			
6487	BE0006	LDX	OBJTBL,Y	: SEE WHAT IT WAS FOR SCORE
648A	A900	LDA	#0	
648C	208973	JSR	ADDSCR	: ADD TO SCORE
648F	A900	LDA	#0	: RESET RPTCNT
6491	990606	STA	OBJTBL+6,Y	
6494	A907	LDA	#7	
6496	200E6D	JSR	SETPH	: SET OBJECT ON EXPLOSION PATH (PATH7)
6499	A41F	LDY	\$1F	: SET BUCK SHOT INDEX
649B	A900	LDA	#0	: RESET RPTCNT
649D	990606	STA	OBJTBL+6,Y	
64A0	A908	LDA	#8	: SET ON CANCEL PATH
64A2	200E6D	JSR	SETPH	
64A5	A5AC	LDA	SYSTAT	: DON'T DEC ON ZORBA LEVELS
64A7	2902	AND	#700000010	
64A9	E012 ^64BD	BEQ	DECI	
64AB	C6A6	DEC	UEOCNT	
64AD	A5A6	LDA	UEOCNT	
64AE	D010 ^64C1	BNE	COL114	
64B1	A902	LDA	#2	
64B3	85A6	STA	UEOCNT	
64B5	A5AC	LDA	SYSTAT	
64B7	0904	ORA	#7000000100	
64B9	29F7	AND	#711110111	



COLLISION DETECT ROUTINE

64BB	85AC	STA	SYSTAT	
64BD	6AA6	DECIT	DEC	UFGCNT
64BE	E00B	64CC	REG	COL115
64C1		COL114		
64C1	208C60	JSR	UEODSP	: DISPLAY UFO'S LEFT
64C4	A900	LDA	#0	: START EXPLOSION SOUND
64C6	20A863	JSR	SNDINI	
64C9	4C7564	JMP	COL160	: NEXT
				: PLAY LEVEL OVER - SET UP PARAMETERS TO START NEXT ONE.
64CC		LVI OVR		
64CC		COL115		
64CC	E6AE	INC	FLYLV	: BUMP PLAY LEVEL
				: ----> DO SOMETHING FANCY TO SHOW PROGRESSION TO NEXT LEVEL
64CE	A900	WARP	LDA	#0
64D0	A207		LIX	#7
64D2	9D00E8	WARPP5	STA	AUDF1,X
64D5	CA		DEX	
64D6	D0FA	64D2	BNE	WARPP5
64D8	A98C		LDA	#8C
64DA	8D01E8		STA	AUDC1
64DD	A98A		LDA	#8A
64DF	8D03E8		STA	AUDC2
64E2	A0F4		LIX	#E4
64E4	AD0AE8	WARP1	LDA	RANDOM
64E7	8D1109		STA	COLOR0
64EA	AD0AE8		LDA	RANDOM
64ED	8D1209		STA	COLOR1
64F0	AD0AE8		LDA	RANDOM
64F3	8D1309		STA	COLOR2
64F6	AD0AE8		LDA	RANDOM
64F9	8D1409		STA	COLOR3
64FC	AD0AE8		LDA	RANDOM
64FF	8D1509		STA	COLOR4
6502	A220		LIX	#20
6504	AD0BD4	WAITVB	LDA	VCOUNT
6507	D0FB	6504	BNE	WAITVB
6509	8C00E8		STY	AUDF1
650C	98		TYA	
650D	49FF		EOR	#FF
650F	8D02E8		STA	AUDF2
6512	CA		DEX	
6513	D0FE	6504	BNE	WAITVB
6515	98		TYA	
6516	48		PHA	
6517	8A		TXA	
6518	48		PHA	
6519	AEOAE8	CHKVAL	LIX	RANDOM
651C	E005		CPX	#5
651E	B0F9	6519	BCS	CHKVAL
6520	209E6D		JSR	UDSCOL
6523	68		PLA	
6524	AA		TAX	
6525	68		PLA	
6526	A8		TAY	
6527	88		DEY	

```

6528 88      DEV
6529 88      DEV
652A 88      DEV
652B B0B7 ^64E4 BNE WARP1
652D 4C8569    JMP     NEWLVL      ; GO TO NEW LEVEL

; CHECK FOR BUCK SHIP COLLIDED WITH POLES, SAUCERS, ETC.
; LOOKS FOR OBJECTS OF SIZE 0. IF SIZE ONE FOUND, THEN CHECK
; IS MADE TO SEE IF OBJECT DESCRIPTOR LEGITIMATE (PC TYPE <> 0).
; IF IT IS, THEN THE OBJECT CAN'T BE A BUCK SHOT, IF ALL
; OF THAT IS OK, THEN A CHECK IS MADE FOR ACTUAL OBJECT COLLISION.

6530      COL120
6530 A59A    LDA     BUCKX      ; SET UP BUCK SHIP PARAMETERS
6532 8512    STA     #12
6534 A59B    LDA     BUCKY
6536 8513    STA     #13
6538 A90B    LDA     #11      ; BUCK LENGTH CONSTANT
653A 8510    STA     #10
653C A914    LDA     #20      ; BUCK WIDTH CONSTANT
653E 8511    STA     #11
6540 A000    LDY     #0       ; START FROM HEAD OF OBJTBL

6542      COL130
6542 B90306  LDA     OBJTBL+3,Y ; SEE IF SIZE=0
6545 F00A ^6551 BEQ     COL150  ; YEP

6547      COL140
6547 98      TYA
6548 18      CLC
6549 6908    ADC     #8
654B A8      TAY
654C C000    CPY     #$00
654E D0F2 ^6542 BNE     COL130
6550      COL145
6550 60      RTS
6551      COL150
6551 B90006  LDA     OBJTBL,Y ; SEE IF OBJECT DESCRIPTOR LEGITIMATE
6554 F0E1 ^6547 BEQ     COL140  ; NO - FORGET IT
6556 C905    CMP     #5       ; CAN'T BE BUCK SHOT
6558 F0ED ^6547 BEQ     COL140  ; CAN'T BE EXPLOSION
655A C907    CMP     #7
655C F0E9 ^6547 BEQ     COL140  ; CAN'T COLLIDE W/ STARS
655E C909    CMP     #9
6560 F0E5 ^6547 BEQ     COL140
6562 20AD66 JSR     CKCRSH    ; SEE IF CRASHED
6565 F0E0 ^6547 BEQ     COL140  ; NOPE

; THIS GUY DIED. - START BUCK EXPLOSION

6567      DEADBK
6567 207F6E JSR     MOVE      ; IN CASE ROOM IS NEEDED
656A A207    LDX     #7       ; BUCK EXPLOSION IS TYPE 7
656C 20E06C JSR     CREATE     ; ALWAYS WORKS
656E D0F6 ^6567 BNE     DEADBK  ; MOVE THESE GUYS OFF AND MAKE ROOM
6571 A52A    LDA     BUCKX
6573 990106 STA     OBJTBL+1,Y
6576 A59B    LDA     BUCKY
6578 990206 STA     OBJTBL+2,Y
657B A909    LDA     #9       ; SET BUCK EXPLOSION PATH
657D 200E6D JSR     SETPTH
6580 A900    LDA     #0       ; RESET MTNSPD

```

COLLISION DETECT ROUTINE

```

6582 8596          STA      MTNSPD
6584 8599          STA      SCORED
6586 20A60         JSR      BDNVSP          ; SET DELAY COUNT
6589 A932          LDA      #50
658B 856A          STA      BDEADC
658D A902          LDA      #2              ; BUCK EXPLOSION SOUND
658F 20A863        JSR      SNDINI
6592 203272        JSR      BSERAS        ; ERASE BUCK SHIP/SHADOW
6595              COL160
6595 207E6E        JSR      MOVE          ; JUST CYCLE EXPLOSION/SOUND
6598 206163        JSR      SOUND
659B 20DB63        JSR      SCORE        ; UPDATE SCORE AS REQ
659E A210          LDY      #510         ; DELAY
65A0 A000          LDY      #0
65A2              COL170
65A2 8B           DEY
65A3 D0ED ^65A2    BNE      COL170
65A5 CA           DEX
65A6 D0EA ^65A2    BNE      COL170
65A8 C66A         DEC      BDEADC
65AA D0E9 ^6595    BNE      COL160
65AC C6AD         DEC      SHPLET
65AE E014 ^65C4    BEQ      COL185        ; GAME OVER
65B0 A59C          LDA      EULAMT        ; SEE IF OUT OF FUEL
65B2 059D          ORA      EULAMT+1
65B4 059E          ORA      EULAMT+2
65B6 D0D9 ^65C1    BNE      COL180        ; NO - KEEP GOING
65B8 A5AE          LDA      PLYLVI        ; JUMP BACK TO FIRST OF 4
65BA 29EC          AND      #$EC
65BC 85AE          STA      PLYLVI
65BE 4C8569        JMP      NEWLVI
65C1              COL180
65C1 4C0569        JMP      NEWPLY        ; CONTINUE PLAY
65C4              COL185
65C4 A940          LDA      #701000000
65C6 8D0ED4        STA      NMIFN
65C9 A900          LDA      #HIGH SLIST
65CB 8D0109        STA      SDSLTH
65CE A900          LDA      #LOW SLIST
65D0 8D0009        STA      SDSLST
65D3 A9E8          LDA      #$E8
65D5 8D0309        STA      CHBAS
65D8 A900          LDA      #50
65DA 8578          STA      TIMER
65DC 8D1209        STA      COLOR1
65DE 8D1309        STA      COLOR2
65E2 8D1409        STA      COLOR3
65E5 8D1509        STA      COLOR4
65E8 8D01E8        STA      AUDC1
65EB 8D03E8        STA      AUDC2
65EE 8D05E8        STA      AUDC3
65F1 8D07E8        STA      AUDC4
65F4 A578          BORING LDA      TIMER
65F6 8D1109        STA      COLOR0
65F9 C9EE          CMP      #$EF
65FB D0F7 ^65F4    BNE      BORING
65FD 4CE568        JMP      CINIT

```

; COLLISION DETECTED BETWEEN BUCK SHOT AND MOTHER ZORBA.  
 ; COLLISION CAN ONLY BE ACCEPTED IF SMALL CENTER REGION

```

; (REACTOR) DETECTS THE COLLISION. OTHERWISE THE
; SHOT IS CANCELED, AND AN ENEMY SHOT IS STARTED IN ITS PLACE
; RETURNING TAKING ONE OF THE STAR PATHS (RANDOM FOR RANDOM
; RICOCHET). IF COLLISION IS DETERMINED ZORBA BLOWS UP AND
; THE ROUND IS BROUGHT TO COMPLETION. MTRHTB CONTAINS PARAMETERS
; FOR FINDING REACTOR GIVEN THE DIFFERENT SIZES OF ZORBAS.

```

```

6600 COL190
; LDA OBJTBL+7,Y ; SEE IF ZORBA ATTACKING
; AND #410
; BNE COL200 ; ALWAYS RETURN REBOUN SHOT HERE
6601 BE0306 LDX OBJTBL+3,Y ; GET ZORBA SIZE INDEX
6602 E006 CPX #6 ; 6,7,8 ARE NO HITS.
6603 B015 ^661C BCS COL200
6604 B0A466 LDA MTRHTB,X ; GET X OFFSET FOR CENTER OF ZORBA
6605 18 CLC
6606 790106 ADC OBJTBL+1,Y ; ADD TO X POSITION FOR TEST
6607 38 SEC
6608 E512 SBC #12 ; GET DISTANCE FROM CENTER
6609 1005 ^6618 BPL COL195 ; GET ABS VALUE FOR DISTANCE
6610 49FF EOR #FF
6611 18 CLC
6612 6901 ADC #1
6613 COL195
6614 C904 CMP #4 ; ACCURACY REQUIREMENT HERE^^^
6615 9027 ^6643 BCC COL220 ; COLLISION DETECTED - ZORBA'S GONE

```

```

; NO COLLISION DETECTED - REDEFINE SHOT AS ENEMY SHOT ON A REBOUND.
; AND LEAVE ZORBA ALONE.

```

```

661C COL200
661D A41F LDY #1F ; POINT TO BUCK SHOT TABLE
661E A900 LDA #0 ; RESET RPTCNT
661F 990606 STA OBJTBL+6,Y
6620 A218 LDA #18 ; SLOW DOWN SHOT TO MIN
6621 990506 STA OBJTBL+5,Y
6622 COL210
6623 AD0AE8 LDA RANDOM ; GET RANDOM PATH 31-38
6624 2907 AND #7
6625 18 CLC
6626 621F ADC #31
6627 200E6D JSR SEIPTH
6628 B90706 LDA OBJTBL+7,Y ; SET BIT 4=1 FOR COLOR
6629 0910 ORA #10
6630 990706 STA OBJTBL+7,Y
6631 A905 LDA #5 ; RICOCHET SOUND
6632 20A863 JSR SNDINI
6633 4C7564 JMP COL160 ; NEXT...

```

```

; ZORBA GOT IT

```

```

6643 COL220

```

```

; eee--->

```

```

; DO SOMETHING FANCY TO SHOW ZORBA BEAFED

```

```
6645 A900 LDA #0
6647 8D00E8 GL0RD STA AUDF1,X
664A CA DEY
664B D0EA 6647 RNE GL0RD
664D A9B0 LDA #B0
664E 85AC STA SYSTAT
6651 A98A LDA #8A
6653 8D01E8 STA AUDC1
6656 8D05E8 STA AUDC9
6659 A00A BOOML LDY #A
665B AD0BD4 BOOZ LDA VCOUNT
665E D0F9 6659 BNE BOOML
6660 88 DEY
6661 D0F8 665B BNE BOOZ
6663 AD0AE8 LDA RANDOM
6666 8D1209 STA COLOR1
6669 AD0AE8 LDA RANDOM
666C 8D1309 STA COLOR2
666E AD0AE8 LDA RANDOM
6672 8D1409 STA COLOR3
6675 AD0AE8 LDA RANDOM
6678 8D1109 STA COLOR0
667B AD0AE8 LDA RANDOM
667E 0908 ORA #200001000
6680 8D1509 STA COLOR4
6683 A5AC LDA SYSTAT
6685 8D04E8 STA AUDF3
6688 AA TAX
6689 BD7775 LDA ZORBD,X
668C 8D00E8 STA AUDF1
668E C6AC DEC SYSTAT
6691 D0D6 6659 BNE BOOML
; THIS PLAY LEVEL OVER - GO TO NEXT
;
6693 COL260
6695 A014 LDY #20 # ADD 20000 PTS (1000*20)
6695 COL280
6695 A210 LDY #10
6697 A900 LDA #0
6699 208973 JSR ADDSCR
669C 88 DEY
669D D0F6 6695 BNE COL280
669F E6AE INC PLAYVL ; GO TO NEXT LEVEL
66A1 4C8569 JMP NEWVI
;
; MOTHER ZORBA HIT TABLE CONTAINS X OFFSET TO CENTER FOR REACTOR BLAST
;
66A4 MTRHTB
66A4 12 DB 18 ; ZORBA SIZE 0
66A5 12 DB 18 ; ZORBA SIZE 1
66A6 0E DB 14 ; ZORBA SIZE 2
66A7 0B DB 11 ; ZORBA SIZE 3
66A8 08 DB 8 ; ZORBA SIZE 4
66A9 07 DB 7 ; ZORBA SIZE 5
66AA 05 DB 5 ; ZORBA SIZE 6
66AB 04 DB 4 ; ZORBA SIZE 7
66AC 02 DB 2 ; ZORBA SIZE 8
```



ROUTINE CHECKS TO SEE IF BASE OBJECT (BUCK OR BUCK SHOT)  
HAS COINCIDENCE W/ OBJECT INDEXED BY Y.

ENTER:

Y=INDEX INTO OBJTBL TO TARGET

\$10=BASE OBJECT LENGTH

\$11=BASE OBJECT WIDTH

\$12=BASE OBJECT X LOCATION

\$13=BASE OBJECT Y LOCATION

\$0B=TARGET OBJECT LENGTH

\$0C=TARGET OBJECT WIDTH

\$0D=DESTROYED

\$0E=DESTROYED

EXIT:

X DESTROYED

\$00-\$02 = USED

\$1E-\$1F = Y SAVE

\$0B-\$0E = USED

IF A=00, NO COLLISION WAS DETECTED

IF A=FF, COLLISION DETECTED

CKCRSH

66AD 841E

STY

\$1E

SAVE TARGET INDEX

66AF B9000A

LDA

OBJTBL,Y

GET PC TYPE OF TARGET

66B2 0A

ASL

A

TYPE\*2 TO X FOR PCSCAL INDEX

66B3 AA

TAX

66B4 20504E

JSR

PCSCAL

GET OBJECT PARAMETERS

66B7 A41E

LDY

\$1E

RESTORE INDEX

66B9 B90206

LDA

OBJTBL+2,Y

GET dY (TARGET Y)-(BASE Y)

66BC 38

SEC

66BD E513

SBC

\$13

66BF B00B ^66CC

BCS

CKCR10

POSITIVE DISTANCE

66C1 49FF

EOR

#\$FF

FORCE POSITIVE DISTANCE

66C3 8901

ADC

#1

C=0

66C5 C50B

CMP

\$B

COMPARE W/ OBJECT LENGTH FOR -dY

66C7 B021 ^66EA

BCS

CKCR50

NO COLLISION

66C9 4CD066

JMP

CKCR20

POSSIBLE COLLISION

66CC

CKCR10

66CC C510

CMP

\$10

COMPARE W/ BUCK SHOT LENGTH FOR +dY

66CE B01A ^66EA

BCS

CKCR50

NO COLLISION

66D0

CKCR20

66D0 B90106

LDA

OBJTBL+1,Y

GET dX (TARGET X)-(BASE X)

66D3 38

SEC

66D4 E512

SBC

\$12

66D6 B00B ^66E3

BCS

CKCR30

POSITIVE DISTANCE

66D8 49FF

EOR

#\$FF

FORCE POSITIVE

66DA 8901

ADC

#1

C=0

66DC C50B

CMP

\$C

COMPARE W/ OBJECT WIDTH FOR -dX

66DE B00A ^66EA

BCS

CKCR50

NO COLLISION

66E0 4CE766

JMP

CKCR40

COLLISION

66E3

CKCR30

66E3 C511

CMP

\$11

COMPARE W/ BUCK SHOT WIDTH FOR +dX

66E5 B003 ^66EA

BCS

CKCR50

NO COLLISION

66E7

CKCR40

66E7 A9FF

LDA

#\$FF

COLLISION DETECTED

66E9 60

RTS



SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 27  
COINCIDENCE CHECKER J:5200 .A65

66EA CKCR50

66EA A900 LDA #0 NO COLLISION

66E9 60 RTS

; ROUTINE INITIALIZES POLE OBJECT DESCRIPTORS AT PERIODIC INTERVALS

```

; POLES
66FD      LDA      SYSTAT      ; SEE IF POLES THIS ROUND
66FE      AND      #2
66F1      BNE      POLE10      ; NO
66F3      LDA      POLCNT      ; SEE IF OK TO PLACE POLES
66F5      BNE      POLE20      ; NO
66F7      LDX      #1          ; SEE IF ROOM FOR 2 POLES
66F9      JSR      CREATE
66FC      BNE      POLE10      ; NO ROOM
66FE      STY      0          ; SAVE POINTER TO 1ST
6700      LDX      #1
6702      JSR      CREATE
6705      BEQ      POLE30      ; GOT 1EM
6707      LDY      0          ; NO ROOM FOR 2ND - KILL FIRST
6709      LDA      #0
670B      STA      OBJTBL,Y
670E      POLE10
670E      RTS
670F      POLE20
670F      DEC      POLCNT      ; DEC COUNTER
6711      JMP      POLE10

; GOT BOTH OBJECT DESCRIPTOR BASES - SET THEM UP
6714      POLE30
6714      STY      1          ; SAVE 2ND DESCRIPTOR INDEX
6716      POLE35
6716      POLE40
6716      LDA      RANDOM      ; GET RANDOM X
6719      CMP      #180        ; RIGHT POLE MUST BE LESS THAN 180
671B      BCC      POLE40
671D      LDY      1
671E      STA      OBJTBL+1,Y  ; SET RIGHT X (Y ALREADY 0)
6722      LDY      0
6724      SEC
6725      SBC      SPREAD      ; DISTANCE BETWEEN POLES
6727      BCC      POLE40      ; CAN'T PUT IT HERE
6729      CMP      #40         ; LEFT MUST BE >50
672B      BCC      POLE40      ; TRY AGAIN
672D      STA      OBJTBL+1,Y
6730      LDA      #8          ; SET AS LEFT POLE
6732      STA      OBJTBL+7,Y
6735      LDA      #9          ; SET SIZE=9
6737      STA      OBJTBL+3,Y
673A      LDY      1
673C      STA      OBJTBL+3,Y
673F      LDA      #20         ; SPEED ALWAYS SAME TO START
6741      STA      OBJTBL+5,Y
6744      LDY      0
6746      STA      OBJTBL+5,Y
6749      LDA      #1          ; SET LEFT POLE AS PATH1
674B      JSR      SETPTH
674E      LDY      1          ; SET RIGHT POLE AS PATH2
6750      LDA      #2
6752      JSR      SETPTH
6755      LDA      POLDLY      ; RESET DELAY COUNTER
6757      STA      POLCNT

```

6755 A5A0  
6757 8582

LDA  
STA  
POLDLY  
POLCNT

RESET DELAY COUNTER

POLE PROCESSING

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 29  
J:5200 .A65

6759 60

RTS

675A PAINT

1	675A	A51D	LDA	#1D	
2	675C	38	SEC		
3	675D	E92D	SBC	#32	
4	675E	A8	TAY		
5	6760	A61E	LDX	#1E	
6	6762	8679	STX	YCOORD	
7	6764	B9157D	LDA	Q4,Y	
8	6767	8570	STA	BYTE	
9	6769	B9157E	LDA	R4,Y	
10	676C	856E	STA	BIT	
11	676E	A8	TAY		
12	676F	B16B	LDA	[ADDR1,Y	
13	6771	8576	STA	SHAPE	
14	6773	B16D	LDA	[ADDRM1,Y	
15	6775	8572	STA	MASK	
16	6777	C8	INY		
17	6778	B16B	LDA	[ADDR1,Y	
18	677A	8577	STA	SHAPE+1	
19	677C	B16D	LDA	[ADDRM1,Y	
20	677E	8573	STA	MASK+1	
21	6780	A008	LDY	#8	LOCATION OF SHAPE SIZE
22	6782	B16B	LDA	[ADDR1,Y	
23	6784	A200	LDX	#0	
24			STA	[ADDRB,X1	
25	6786	A679	LDX	YCOORD	
26	6788	857F	STA	WIDTH	
27	678A	18	CLC		
28	678B	2470	BIT	BYTE	
29	678D	102B ^67BA	BPL	RCLIP	
30	678F	6570	ADC	BYTE	
31	6791	902A ^67B9	BCC	EXITLC	
32	6793	902A ^67B9	BEQ	EXITLC	
33	6795	857E	STA	WIDTHC	
34	6797	A57F	LDA	WIDTH	
35	6799	38	SEC		
36	679A	E57E	SBC	WIDTHC	
37	679C	18	CLC		
38	679D	6576	ADC	SHAPE	
39	679F	8576	STA	SHAPE	
40	67A1	9002 ^67A5	BCC	GL1	
41	67A3	E677	INC	SHAPE+1	
42	67A5	A57F GL1	LDA	WIDTH	
43	67A7	38	SEC		
44	67A8	E57E	SBC	WIDTHC	
45	67AA	18	CLC		
46	67AB	6572	ADC	MASK	
47	67AD	8572	STA	MASK	
48	67AF	9002 ^67B3	BCC	GL2	
49	67B1	E673	INC	MASK+1	
50	67B3	A900 GL2	LDA	#0	
51	67B5	8570	STA	BYTE	
52	67B7	F00F ^67C8	BEQ	A1	
53	67B9	60 EXITLC	RTS		
54	67BA	857F RCLIP	STA	WIDTHC	
55	67BC	6570	ADC	BYTE	
56	67BE	C928	CMP	#40	
57	67C0	9006 ^67C8	BCC	A1	
	67C2	A928	LDA	#40	
	67C4	E570	SBC	BYTE	

## PAINT ROUTINE

```
67C6 857E STA WIDTHC
67C8 C8 A1 INY
67C9 B16B LDA [ADDR],Y
67CB 8574 STA HIGHT
67CD A001 LDY ##1
        : STA [ADDRB],Y
        : LDA ADDR8
67CF 18 CLC
67D0 6902 ADC ##2
        : STA BACK
        : LDA ADDR8+1
67D2 6900 ADC ##0
        : STA BACK+1
67D4 E090 HERE CPX #144
67D6 B01F ^67F7 BCS TH
67D8 BDB17E LDA P1H,X
67DB 8575 STA SCREEN+1
67DD BD157E LDA P1L,X
67E0 18 CLC
67E1 A570 ADC BYTE
67E3 8574 STA SCREEN
67E5 9002 ^67E9 BCC OKP1
67E7 E675 INC SCREEN+1
67E9 A47E OKP1 LDY WIDTHC
67EB 88 DEY
67ED R174 DRAW LDA [SCREEN],Y
67EE 3172 AND [MASK],Y ; <-----BUG?
67F0 1176 ORA [SHAPE],Y ; <-----" " "
67F2 9174 STA [SCREEN],Y
67F4 88 DEY
67F5 10E5 ^67EC BPL DRAW
67F7 18 TH CLC
67F8 A57F LDA WIDTH
67FA 6572 ADC MASK
67FC 8572 STA MASK
67FE 9002 ^6802 BCC GL3
6800 E673 INC MASK+1
6802 A57F GL3 LDA WIDTH
6804 18 CLC
        : ADC BACK
        : STA BACK
6805 9000 ^6807 BCC GL4
        : INC BACK+1
6807 A57F GL4 LDA WIDTH
6809 18 CLC
680A 6576 ADC SHAPE
680C 8576 STA SHAPE
680E 9002 ^6812 BCC GL5
6810 E677 INC SHAPE+1
6812 E8 GL5 INX
6813 C671 DEC HIGHT
6815 D0RD ^67D4 BNE HERE
6817 60 RTS
```



; ROUTINE ERASES AN OBJECT FROM VIDRAM

; ENTER:

; 1C = PICTURE NUMBER TO DRAW

; 1D = X DISPLAY POSITION

; 1E = Y DISPLAY POSITION

; 1F = FG COLOR

; 00-01 = USED

; 05-07 = USED

; 10-1B = USED

; 1C-1C = CHANGED

```

6818      ERASE
6818 A51D      LDA      $1D
681A 38        SEC
681B E920      SBC      #32
681D A8        TAY
681E A41E      LDX      $1E
6820 8679      STX      YCOORD
6822 B9157D    LDA      Q4,Y
6825 8570      STA      BYTE
6827 B9157E    LDA      R4,Y
682A 856F      STA      BIT
682C A8        TAY
682D B16B      LDA      [ADDR],Y
682F 8576      STA      SHAPE
6831 B16D      LDA      [ADDRM],Y
6833 8572      STA      MASK
6835 C8        INY
6836 B16B      LDA      [ADDR],Y
6838 8577      STA      SHAPE+1
683A B16D      LDA      [ADDRM],Y
683C 8573      STA      MASK+1
683E A008      LDY      #8          ;LOCATION OF SHAPE SIZE
6840 B16B      LDA      [ADDR],Y
6842 A200      LDX      #0
6844          STA      [ADDRB,X1]
6844 A679      LDX      YCOORD
6846 857F      STA      WIDTH
6848 18        CLC
6849 2470      BIT      BYTE
684B 102B ^6878 RPL      ZRCLIP
684D 6570      ADC      BYTE
684E 9026 ^6877 BCC      ZEXITLC
6851 F024 ^6877 BEQ      ZEXITLC
6853 857E      STA      WIDTHC
6855 A57F      LDA      WIDTH
6857 38        SEC
6858 E57E      SBC      WIDTHC
685A 18        CLC
685B 6576      ADC      SHAPE
685D 8576      STA      SHAPE
685F 9002 ^6863 BCC      7GI 1
6861 E677      INC      SHAPE+1
6863 A57F      LDA      WIDTH
6865 38        SEC
```

6861 E677 INC SHAPE+1  
6863 A57F ZGL1 LDA WIDTH  
6865 38 SEC

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 33  
ERASE OBJECT ROUTINE J:5200 .A65

6866 E57E SBC WIDTHC  
6868 18 CLC  
6869 6572 ADC MASK  
686B 8572 STA MASK  
686D 9002 ^6871 BCC ZGL2  
686E E673 INC MASK+1  
6871 A900 ZGL2 LDA #0  
6873 8570 STA BYTE  
6875 F00F ^6886 BEQ ZA1  
6877 60 ZEXITLC RTS  
6878 857E ZRCLIP STA WIDTHC  
687A 6570 ADC BYTE  
687C C928 CMP #40  
687E 9006 ^6886 BCC ZA1  
6880 A928 LDA #40  
6882 E570 SBC BYTE  
6884 857E STA WIDTHC  
6886 C8 ZA1 INY  
6887 B16B LDA [ADDR1,Y  
6889 8571 STA RIGHT  
688B A579 LDA YCOORD  
688D 4A LSR A  
688E 18 CLC  
688F 690A ADC #10  
6891 C970 CMP #70  
6893 9002 ^6897 BCC ERGTIT  
6895 A970 LDA #70  
6897 C0BD4 ERGTIT CMP VCOUNT  
689A B0FB ^6897 BCS ERGTIT  
689C A001 LDY #1  
689E 18 CLC  
689F 6902 ADC #2  
68A1 6900 ADC #0  
68A3 E090 ZHERE CPX #144  
68A5 B01D ^68C4 BCS ZTH  
68A7 BDB17E LDA P1H,X  
68AA 8575 STA SCREEN+1  
68AC BD157F LDA P1L,X  
68AF 18 CLC  
68B0 6570 ADC BYTE  
68B2 8574 STA SCREEN  
68B4 9002 ^68BB BCC ZOKP1  
68B6 E675 INC SCREEN+1  
68B8 A47E ZOKP1 LDY WIDTHC  
68BA 88 DEY  
68BB B174 ZDRAW LDA [SCREEN],Y  
68BD 3172 AND [MASK],Y ;<-----BUG?  
68BF 2174 STA [SCREEN],Y  
68C1 88 DEY  
68C2 10F7 ^68BB BPL ZDRAW  
68C4 18 ZTH CLC  
68C5 A57F LDA WIDTH  
68C7 6572 ADC MASK  
68C9 8572 STA MASK  
68CB 9002 ^68CF BCC ZGL3  
68CD E673 INC MASK+1  
68CF A57F ZGL3 LDA WIDTH  
68D1 18 CLC  
; ADC BACK  
; STA BACK

ERASE OBJECT ROUTINE

J:5200 .A65

68D2	9000 ^68D4	BCC	ZGL4
68D4	A57F	INC	BACK+1
68D6	18	LDA	WIDTH
68D7	6576	ELC	
68D9	8576	ADC	SHAPE
68DB	9002 ^68DF	STA	SHAPE
68DD	E677	BCC	ZGL5
68DF	E8	INC	SHAPE+1
68E0	C671	INX	
68E2	D0BF ^68A3	DEC	HIGHT
68E4	6D	BNE	ZHERE
		RIS	

CONTROL REGISTERS INITIALIZED AS FOLLOWS:

68E5 CINIT

68E5 78 SEI  
68E6 D8 CLD  
68E7 A900 LDA #500  
68E9 8D0ED4 STA NMIFN ; DON'T BE RUDE!! (AND INTERRUPT)  
68EC AA TAX

68ED ZAPIT

68ED 9D00C0 STA \$C000,X  
68E0 9D00D4 STA \$D400,X  
68E3 9D00E8 STA \$E800,X  
68E6 9D00F3 STA \$F300,X  
68E9 9D0006 STA \$600,X  
68EC 9D0002 STA \$200,X  
68FF 9500 STA \$00,X  
6901 E8 INX  
6902 D0E9 ^68ED BNE ZAPIT

ENTRANCE FOR RESTART

6904 INIT

6904 A940 LDA #01000000  
6906 8D0EE8 STA \$E80E  
6909 A900 LDA #0  
690B 8D00D4 STA \$D400  
690E 8D1DC0 STA GRCTL  
6911 A9B1 LDA #LOW DLILOC ; INIT DLI VECTOR  
6913 8D0602 STA VDSLST  
6916 A900 LDA #HIGH DLILOC  
6918 8D0702 STA VDSLST+1  
691B A9DE LDA #LOW VBIRTN ; INIT VBI VECTOR  
691D 8D0202 STA VVBRTI  
6920 A973 LDA #HIGH VBIRTN  
6922 8D0302 STA VVBRTI+1  
6925 A902 LDA #00000010 ; INVERSE CHARS  
6927 8D0409 STA CHART  
692A A93C LDA #00111100 ; SETUP PIA FOR JOYSTICK READ  
692C 8D02D3 STA PACTI  
692F 8D03D3 STA PBCTL  
6932 A902 LDA #00000010 ; SETUP POKEY FOR KEYBOARD READ AND DEBOUNCE  
6934 8D0FE8 STA \$E80E  
6937 8D0FE8 STA SKCTL  
693A 206573 JSR CHKHSC ; CHECK HIGH SCORE  
693D 206371 JSR TITLE ; DISPLAY TITLE PAGE  
6940 A977 LDA #HIGH MOUDAT ; MUST BE ON A 2K BOUNDARY  
6942 8D0309 STA CHBAS  
6945 A900 LDA #01000000  
6947 8D0ED4 STA NMIFN  
694A A941 LDA #LOW DLIST ; INIT DISPLAY LIST (CANNOT CROSS 1K BOUND.)  
694C 8D0009 STA SDLSTL  
694F A980 LDA #HIGH DLIST  
6951 8D0109 STA SPLSTH

; COPY DLI INTO ZERO PAGE

```

6954 A240      LDX      #40
6956 BDER80    CLOOP    LDA      BEFL0C,X
6959 95B1      STA      BLILOE,X
695B CA        DEX
695C 10F8 ^6956 BPL      CLOOP
695E A202      LDA      #2
6960 857D      STA      SPEEDC
6962 8DCA00    STA      SPEED
;-----
6965 A205      LDA      #5      ; FIRE DELAY AND FUDGE AMOUNT
6967 85AF      STA      FIRDLY
6969 85AD      STA      SHPLFT
696B A208      LDA      #8
696D 85B0      STA      FIRFDG

```

; INITIALIZE VARIABLES THAT ARE DONE ONLY AT BEGINNING OF GAME.

; DATA DERIVED FROM TABLE

```

696E A200      LDA      #0      ; STARTING PLAY LEVEL =0
6971 857B      STA      LVLFLG
6973 85AC      STA      SYSTAT
6975 85AE      STA      PLYLVL
6977 857C      STA      MEANFG
6979 8D0D02    STA      CSCORE      ; ZERO OUT SCORE
697C 8D0E09    STA      CSCORE+1
697F 8D0F09    STA      CSCORE+2
6982 8D1009    STA      CSCORE+3

```

; DATA TO BE INITIALIZED BEFORE EACH PLAY LEVEL

```

6985 NEWLVL
6985 A5AE      LDA      PLYLVL      ; GET CURRENT PLAY LEVEL
6987 E914      CMP      #MAXLVL    ; CHECK FOR MAXIMUM LEVEL
6989 900C ^6997 BCC      NEWL05
698B 38        SEC
698C E904      SRC      #4      ; IF PAST MAXIMUM, GO BACK 4
698E 85AE      STA      PLYLVL
6990 A57C      LDA      MEANFG
6992 18        CLC
6993 A205      ADC      #5
6995 857C      STA      MEANFG
6997 NEWL05

```

```

U 6997 200000  JSR      LEVELD
699A A5AE      LDA      PLYLVL
699C A20F      LDX      #PTCNT    ; #NUMBER OF DATA BYTES TO INITIALIZE
699E 20C273    JSR      MULT
69A1 8501      STA      1
69A3 8600      STX      0
69A5 A925      LDA      #LOW PLYTBL ; OFFSET TO BASE OF TABLE
69A7 18        CLC
69A8 6500      ADC      0
69AA 8500      STA      0
69AC A96A      LDA      #HIGH PLYTBL
69AE 6501      ADC      1
69B0 8501      STA      1
69B2 A000      LDY      #0      ; INIT INDEX
69B4 NEWL10
69B4 B100      LDA      [0],Y
69B6 999F00    STA      LVLDAT,Y

```



```

69B9 C8      INY
69BA E00E    CRY
69BC D0E6    BNE      NEWL10
; DON'T LET THEM PLAY FOREVER

69BE A5A7    LDA      MINSPI
69C0 18      CLC
69C1 657C    ADC      MEANFG
69C3 85A7    STA      MINSPI

; SEE IF TO GIVE NEW TANK OF FUEL
; ONLY GIVE NEW FUEL WHEN STARTING 1ST OF 4

69C5 A5AE    LDA      PLYLVL
69C7 2903    AND      #3
69C9 D00A    BNE      NEWL20
69CB A960    LDA      #96
69CD 859E    STA      FULAMT+2
69CF A900    LDA      #0
69D1 859C    STA      FULAMT
69D3 859D    STA      FULAMT+1
69D5        NEWL20
; DATA TO BE INITIALIZED BEFORE EACH PLAY
; NEWPLY
69D5        LDA      #47
69D7 A92E    STA      MISRT
69D9 8597    LDA      #0
69DB A900    STA      MTNCNT
69DD 8598    JSR      BDINIT
69DF 20256D  JSR      SETCOL
69E1 20926D  LDA      SYSTAT
69E3 A5AC    AND      #700000010
69E5 2902    BNE      SKPDL1
69E7 D003    JSR      SETDL1
69E9 20826D  LDA      #96
69EB A960    STA      BUCKY
69ED 859B    LDA      #112
69EF A970    STA      BUCKX
69F2 859A    JSR      SHIPS
69F4 20B26D  LDA      MINSPI
69F7 A5A7    STA      SCRSPD
69F9 8599    JSR      CONVSP
69FB 206A60  LDA      #0
69FD A900    STA      MTNSPI
6A00 8596    STA      AUDC1
6A02 8D01E8  STA      AUDC2
6A05 8D03E8  STA      AUDC3
6A08 8D05E8  STA      AUDC4
6A0B 8D07E8  STA      SDSTAT
6A0E 8589    TAY
6A10 A8      NEWL10
6A11        LDA      #0
6A11 A900    STA      OBJTBL,Y
6A13 990006  INY
6A16 C8      BNE      NEWL10
6A17 D0E8    NEWL10

```

; ONLY GIVE NEW FUEL WHEN STARTING 1ST OF 4

; NEW TANK - START WITH AMOUNT DISPLAYABLE

; INITIALIZE BACKDROP

; SET BUCK START LOCATION

; CENTER

; FORCE INITIAL SHIP DISPLAY

; TURN OF SOUND GENERATOR

: ---> DISPLAY INITIAL FUEL GRAPH (JSR FUEL10) AND INITIAL  
UFDCNT (JSR UFODSP)

6A19 200163	JSR	FUEL10	
6A1C 208C60	JSR	UFODSP	
6A1E A2EE	LDX	#\$FF	: INIT STACK
6A21 9A	TXS		
6A22 4C2B60	JMP	DRIVER	: DONE - NOW RUN PROGRAM

PLAY LEVELS TABLE, CONTAINS THE DIFFICULTY PARAMETERS FOR THE  
VARIOUS LEVELS OF PLAY

BYTES ARE ORDERED AS FOLLOWS:

RSPONS JOY STICK RESPONSE  
POLDLY NUMBER OF PASSES BETWEEN POLE SETS  
SAUDLY NUMBER OF PASSES BETWEEN SAUCER SETS  
SAUSPD BASE SPEED OF SAUCERS  
HOPDLY NUMBER OF PASSES BETWEEN HOPPER APPEARANCES  
HOPSPD BASE SPEED OF HOPPERS  
MTRSPD MOTHER ZORBA SPEED  
LFOCNT NUMBER OF UFOS (POLES) TO KILL  
MINSPD MINIMUM SPEED BUCK CAN TRAVEL  
SPREAD DISTANCE BETWEEN POLES  
HERDLY HOPPER FIRE DELAY  
PERDLY POLE FIRE DELAY  
MERDLY MOTHER ZORBA FIRE DELAY  
SYSTAT SYSTEM PLAY LEVEL STATUS BITS  
7 = HOPPERS FIRE  
6 = POLES FIRE IN  
5 = POLES FIRE OUT  
4 = HOPPERS EXIST  
3 = SAUCERS EXIST  
2 = MOTHER ZORBA  
1 = SPACE SCENE  
0 =

6A25

PLYTBL

			RSP	PDL	SDL	SSP	HDL	HSP	MSF	UFO	MIN	SPR	HFD	PFD	MED	SYS
6A25	0420000000	DB	\$04	\$20	\$00	\$00	\$00	\$00	\$00	\$0A	\$35	\$25	\$00	\$00	\$00	\$00
6A33	041E583000	DB	\$04	\$1E	\$58	\$30	\$00	\$00	\$00	\$0F	\$35	\$25	\$00	\$00	\$00	\$08
6A41	041D583040	DB	\$04	\$1D	\$58	\$30	\$40	\$20	\$00	\$14	\$35	\$25	\$00	\$00	\$00	\$18
6A4F	0400203000	DB	\$04	\$00	\$20	\$30	\$00	\$00	\$60	\$0A	\$35	\$00	\$00	\$00	\$00	\$0A
6A5D	0614000000	DB	\$06	\$14	\$00	\$00	\$00	\$00	\$00	\$0F	\$3C	\$23	\$00	\$20	\$00	\$20
6A6B	0616503800	DB	\$06	\$16	\$50	\$38	\$00	\$00	\$00	\$14	\$3C	\$23	\$00	\$28	\$00	\$28
6A79	0616503835	DB	\$06	\$16	\$50	\$38	\$35	\$30	\$00	\$19	\$3C	\$23	\$38	\$28	\$00	\$B8
6A87	0600203800	DB	\$06	\$00	\$20	\$38	\$00	\$00	\$60	\$0F	\$40	\$00	\$00	\$00	\$00	\$0A
6A95	0610000000	DB	\$06	\$10	\$00	\$00	\$00	\$00	\$00	\$14	\$48	\$20	\$00	\$18	\$00	\$20
6AA3	0612403C20	DB	\$06	\$12	\$40	\$3C	\$20	\$20	\$00	\$19	\$48	\$23	\$00	\$20	\$00	\$38
6AB1	0612403C20	DB	\$06	\$12	\$40	\$3C	\$20	\$30	\$00	\$1E	\$48	\$23	\$28	\$20	\$00	\$B8
6ABF	0600203C00	DB	\$06	\$00	\$20	\$3C	\$00	\$00	\$60	\$14	\$50	\$00	\$00	\$00	\$00	\$0A
6ACD	060C803000	DB	\$06	\$0C	\$80	\$30	\$00	\$00	\$00	\$19	\$50	\$20	\$20	\$10	\$00	\$28
6ADB	060E20301A	DB	\$06	\$0E	\$20	\$30	\$1A	\$40	\$00	\$1E	\$50	\$20	\$18	\$10	\$00	\$B8
6AE9	060E20301A	DB	\$06	\$0E	\$20	\$30	\$1A	\$40	\$00	\$28	\$50	\$20	\$10	\$10	\$00	\$98
6AF7	0600203000	DB	\$06	\$00	\$20	\$30	\$00	\$00	\$60	\$20	\$58	\$00	\$00	\$00	\$00	\$0A
6B05	060A605000	DB	\$06	\$0A	\$60	\$50	\$00	\$00	\$00	\$1E	\$70	\$14	\$00	\$08	\$00	\$28
6B13	060B185018	DB	\$06	\$0B	\$18	\$50	\$18	\$40	\$00	\$23	\$60	\$18	\$10	\$08	\$00	\$B8
6B21	060B185010	DB	\$06	\$0B	\$18	\$50	\$10	\$50	\$00	\$28	\$60	\$18	\$08	\$04	\$00	\$98
6B2F	0600185000	DB	\$06	\$00	\$18	\$50	\$00	\$00	\$70	\$25	\$58	\$00	\$00	\$00	\$00	\$0A

= 0014

MAXLVL

EQU

(\*-PLYTBL)/PTCNT

ROUTINE MAINTAINS VARIABLES ASSOCIATED WITH SCROLLING  
MOUNTAIN RANGE AND CAUSES MOUNTAIN RANGE TO SCROLL

```

5      6B3D      MTNWRK
6      6B3D A59F      LDA      RSPONS      ; GET SLOWDOWN AMOUNT
7      6B3F 4A          LSR          A
8      6B40 D002 ^6B44 BNE      MTNW05
9      6B42 A901      LDA      #1          ; CANT BE 0
10     6B44          MTNW05
11     6B44 8500      STA      0
12     6B46 A560      LDA      JSDATA      ; SEE IF GOING RIGHT
13     6B48 101C ^6B66 BPL      MTNW30      ; YES
14     6B4A 2910      AND      #$10        ; LEFT?
15     6B4C F024 ^6B72 BEQ      MTNW40      ; YES
16
17     6B4E A596      LDA      MTNSPD      ; JOYSTICK UP - SLOW DOWN MOUNTAINS
18     6B50 100A ^6B5C BPL      MTNW10      ; POSITIVE MTNSPD
19     6B52 18          CLC
20     6B53 6500      ADC      0
21     6B55 3024 ^6B7B BMI      MTNW50
22     6B57 A200      LDA      #0          ; STOP AT 0
23     6B59 4C7B6B    JMP      MTNW50
24     6B5C          MTNW10
25     6B5C 38          SEC
26     6B5D E500      SBC      0
27     6B5F 101A ^6B7B BPL      MTNW50
28     6B61 A200      LDA      #0          ; STOP AT 0
29     6B63 4C7B6B    JMP      MTNW50
30
31     6B66          MTNW30
32     6B66 A596      LDA      MTNSPD      ; JOYSTICK RIGHT - DEC MTNSPD
33     6B68 38          SEC
34     6B69 E59F      SBC      RSPONS
35     6B6B 500E ^6B7B BVC      MTNW50
36     6B6D A980      LDA      #-128
37     6B6F 4C7B6B    JMP      MTNW50      ; MIN = -128
38     6B72          MTNW40
39     6B72 A596      LDA      MTNSPD      ; JOYSTICK LEFT - INC MTNSPD
40     6B74 18          CLC
41     6B75 652F      ADC      RSPONS
42     6B77 5002 ^6B7B BVC      MTNW50
43     6B79 A97F      LDA      #127
44     6B7B          MTNW50      ; MAX=127
45     6B7B 8596      STA      MTNSPD
46
47     ; SET UP TO DRAW MOUNTAINS
48
49     6B7D          MTNW60
50     6B7D 18          CLC
51     6B7E A596      LDA      MTNSPD      ; GET ABS(MTNSPD)
52     6B80 1005 ^6B87 BPL      MTNW70
53     6B82 49EF      EOR      #$FF
54     6B84 6901      ADC      #1
55     6B86 18          CLC
56     6B87          MTNW70
57     6B87 4A          LSR          A      ; /2
58     6B88 4A          LSR          A      ; /4
59     6B89 6598      ADC      MTNCNT      ; ADD TO CURRENT COUNT BASE

```

```
6B8B 8598 STA MTNCNT
6B8D 108D ^6BCC BPL MTN130 ; DONT MOVE MOUNTAINS
6B8F 297F AND #7F ; RESET TOP BIT FOR NEXT TIME
6B91 8598 STA MTNCNT
6B93 A596 LDA MTNSPD ; SEE IF TO MOVE LEFT OR RIGHT
6B95 100E ^6BA6 BPL MTNWS0 ; POSITIVE SPEED - DEC MISTRT
6B97 E697 INC MTSTRT ; INC MOUNTAIN START ADDRESS
6B99 A597 LDA MTSTRT ; CHECK ROLLOVER
6B9B C94E CMP #78
6B9D D013 ^6BB2 BNE MTNWS0 ; DRAW NEW MOUNTAIN RANGE
6B9F A921 LDA #33
6BA1 8597 STA MTSTRT
6BA3 4CB26B JMP MTNWS0
6BA6 MTNWS0
6BA6 C697 DEC MTSTRT
6BA8 A597 LDA MTSTRT
6BAA C920 CMP #32
6BAC D004 ^6BB2 BNE MTNWS0 ; DRAW NEW MOUNTAIN RANGE
6BAE A94D LDA #77
6BB0 8597 STA MTSTRT
; DRAW MOUNTAIN RANGE $0260-$0277
6BB2 MTNWS0
6BB2 A5AC LDA SYSTAT
6BB4 2902 AND #%00000010
6BB6 D014 ^6BCC BNE MTN130
6BB8 A000 LDY #00
6BBA A697 LDX MTSTRT
6BBC MTNWS0
6BBC 8A TXA
6BBD 99C820 STA $2008,Y
6BC0 E8 INX
6BC1 E04E CPX #78
6BC3 D002 ^6BC7 BNE M1OK
6BC5 A221 LDX #33
6BC7 C8 M1OK INY
6BC9 C028 CPY #40
6BCA D0F0 ^6BBC BNE MTNWS0
6BCC MTN130
6BCC 60 RTS
```



ROUTINE CREATES 1-4 SAUCERS (RANDOM) STARTING ON THE  
 SAME PATH BUT AT DIFFERENT DISTANCES INTO THE PATH.  
 WON'T CREATE UNLESS COUNT IS SET.

6BCD SAUCER  
 6BCD A5AC LDA SYSTAT ; SEE IF THEY EXISTS HERE  
 6BCF 2908 AND #8  
 6BD1 F006 ^6BD9 BEQ SAUC10 ; NO  
 6BD3 A583 LDA SAUCNT  
 6BD5 F003 ^6BDA BEQ SAUC20 ; CREATE SAUCERS  
 6BD7 C683 DEC SAUCNT ; JUST DEC COUNT AND EXIT

6BD9 SAUC10  
 6BD9 60 RTS  
 6BDA SAUC20  
 6BDA A902 LDA #2 ; SEE HOW MANY SAUCERS TO CREATE (1-4)

6BDC 851F STA #1F ; SAVE COUNTER  
 6BDE AD0AE8 SAUX LDA RANDOM ; GET STARTING X  
 6BE1 297F AND #20111111

6BE3 18 CLC  
 6BE4 6920 ADC #32  
 6BE6 851F STA #1F  
 6BE8 AD0AE8 LDA RANDOM ; GET PATH (3,4,5,6)  
 6BE9 2903 AND #3  
 6BED 18 CLC  
 6BEE 6903 ADC #3  
 6BF0 851D STA #1D

6BF2 SAUC30  
 6BF2 A202 LDX #2 ; TRY TO CREATE A SAUCER  
 6BF4 20E06C JSR CREATE  
 6BF7 D0E0 ^6BD9 BNE SAUC10 ; NO MORE ROOM - EXIT  
 6BF9 A51D LDA #1D ; SET PATH  
 6BFB 200E6D JSR SETPTH  
 6BFE A51F LDA #1F ; SET X  
 6C00 990106 STA OBJTBL+1,Y

SET INITIAL RPTCNT=OBJECT#\*37 (FROM TABLE)  
 Y=RPTCNT+144

6C03 A61F LDX #1F ; GET OBJECT#  
 6C05 BD1F6C LDA RPTBL,X ; GET #\*28  
 6C08 990606 STA OBJTBL+6,Y ; SET RETCNT  
 6C0B 18 CLC ; SET Y  
 6C0C 6990 ADC #144  
 6C0E 990206 STA OBJTBL+2,Y

6C11 A5A2 LDA SAUSPD ; SET SAUCER SPEED  
 6C13 990506 STA OBJTBL+5,Y  
 6C16 C61F DEC #1F

6C18 10D8 ^6BF2 BPL SAUC30  
 6C1A SAUC40  
 6C1A A5A1 LDA SAUDLY ; RESET COUNT

6C1C 8583 STA SAUCNT  
 6C1E 60 RTS

6C1F RPTBL  
 6C1F 00254A6F DB 0,37,74,111

DOES NECESSARY WORK FOR PROCESSING BUCK ROGER SHIP.  
 WHEN JOYSTICK PULLED LEFT OR RIGHT, BS MOVES LEFT OR RIGHT  
 UP TO BOUNDARIES. WHEN JOYSTICK PUSHER UP OR PULLED DOWN,  
 BUCK SHIP MOVES UP OR DOWN UNTIL BOUNDARIES ARE HIT AT  
 WHICH TIME SCRSPD IS INCREASED (TOP BOUNDARY) OR DECREASED  
 (LOWER BOUNDARY). WHEN FIRE BUTTON PRESSED, SHOTS ARE FIRED  
 STARTING FROM BS AND TAKING A SPECIAL PATH DEFINED FOR  
 BUCK SHOTS.  
 BUCK SHIP CAN GO NO LOWER THAN 128, AND NO HIGHER THAN 96.  
 X LIMITS ARE 48CX<176 SHADOW COPIES BUCK X BUT Y IS FIXED  
 AT 131.

BSWORK

BSERAS : ERASE BUCK ROGERS

6C23	203272	JSR	BSERAS	:	ERASE BUCK ROGERS
6C26	A560	LDA	JSDATA	:	MOVE BUCK
6C28	1010 ^6C3A	BPL	BSW030	:	RIGHT
6C2A	2910	AND	#10	:	
6C2C	F010 ^6C4A	BEQ	BSW040	:	LEFT
6C2E	A901	LDA	#1	:	FORCE BUCK CENTER PICTURE
6C30	8580	STA	LASTBS	:	
6C32	4C576C	JMP	BSW050	:	SKIP CENTER FLOAT
		LDA	BUCKX	:	FLOAT BUCK TO CENTER
		CMP	#112	:	SEE IF LEFT OR RIGHT OF CENTER
		BEQ	BSW050	:	AT CENTER - JUST DRAW IT
		BMI	BSW010	:	LEFT OF CENTER
		SEC		:	
		SBC	#2	:	RIGHT OF CENTER - ADJUST X
		JMP	BSW020	:	GO SET X AND DRAW
		BSW010		:	
		CLC		:	
		ADC	#2	:	
6C35		BSW020		:	
6C35	859A	STA	BUCKX	:	
6C37	4C576C	JMP	BSW050	:	NOW DRAW IT
				:	MOVE BUCK SHIP TO RIGHT WITH RIGHT TILT
				:	
6C3A		BSW030		:	
6C3A	A902	LDA	#2	:	FORCE RIGHT TILT
6C3C	8580	STA	LASTBS	:	
6C3E	A59A	LDA	BUCKX	:	ATTEMPT MOVE RT
6C40	18	CLC		:	
6C41	6904	ADC	#4	:	
6C43	C9B1	CMP	#177	:	CHECK RIGHT LIMIT
6C45	B010 ^6C57	BCS	BSW050	:	TOO FAR RIGHT - DONT MOVE IT
6C47	4C356C	JMP	BSW020	:	SET NEW X AND DRAW
				:	MOVE SHIP LEFT W/ LEFT TILT
				:	
6C4A		BSW040		:	
6C4A	A900	LDA	#0	:	FORCE LEFT TILT
6C4C	8580	STA	LASTBS	:	
6C4E	A59A	LDA	BUCKX	:	ATTEMPT MOVE LEFT
6C50	38	SEC		:	
6C51	E904	SBC	#4	:	
6C53	C930	CMP	#48	:	

```
6C55 B0DE ^6C35      BCS      BSW020      ; OK - SET X AND DRAW IT
; POSITION BUCK SHIP BY Y
6C57      BSW050
6C57 A560      LDA      JSDATA      ; SEE IF ATTEMPTING UP OR DOWN
6C59 2904      AND      #4
6C5B F012 ^6C6F      BEQ      BSW060      ; UP
6C5D A560      LDA      JSDATA
6C5E 2908      AND      #8
6C61 D033 ^6C26      BNE      BSW100      ; NEITHER - XY SET - GO DRAW IT
;
; MOVE BS DOWN
6C63 A59B      LDA      BUCKY      ; ATTEMPT MOVE DOWN
6C65 18      CLC
6C66 A903      ADC      #3
6C68 C981      CMP      #129
6C6A B01E ^6C8A      BCS      BSW090      ; TOO LOW - SLOW DOWN SCROLL
6C6C 4C786C      JMP      BSW070      ; OK TO MOVE DOWN
;
; MOVE BS UP
6C6F      BSW060
6C6F A59B      LDA      BUCKY      ; ATTEMPT MOVE UP
6C71 38      SEC
6C72 E903      SBC      #3
6C74 C260      CMP      #96
6C76 2005 ^6C7D      BCC      BSW080      ; TOO HIGH - SPEED UP SCROLL
6C78      BSW070
6C78 859B      STA      BUCKY      ; SET NEW Y POSITION
6C7A 4C966C      JMP      BSW100      ; GO DRAW BS
;
; BUCK SHIP AT UPPER BOUNDARY - DO SPEED INCREASE
6C7D      BSW080
6C7D E699      INC      SCRSPD      ; INC SCROLL SPEED
6C7F A9C0      LDA      #C0
6C81 C599      CMP      SCRSPD
6C83 B011 ^6C96      BCS      BSW100      ; CHECK FF LIMIT
6C85 8599      STA      SCRSPD
6C87 4C966C      JMP      BSW100      ; DRAW BS
;
; BUCK SHIP AT LOWER BOUNDARY - DO SPEED DECREASE
6C8A      BSW090
6C8A C699      DEC      SCRSPD      ; DOWN - DEC SCROLL SPEED
6C8C A599      LDA      SCRSPD      ; CHECK LOWER LIMIT
6C8E C5A7      CMP      MINSPD
6C90 B004 ^6C96      BCS      BSW100      ; OK
6C92 A5A7      LDA      MINSPD      ; SET AS LOWER LIMIT
6C94 8599      STA      SCRSPD
;
; DRAW BUCK SHIP AND SHADOW
6C96      BSW100
6C96 206A60      JSR      CONVSP
6C98 4C6F72      JMP      BSPAIN      ; PAINT BUCK SHIP, SHADOW AND EXIT
```

ROUTINE CHECKS FIRE BUTTON TO SEE IF TO FIRE SHOTS.  
IF BUTTON IS HELD DOWN, A DELAY COUNTER REGULATES THE  
SPEED AT WHICH SHOTS CAN BE FIRED. IF THE BUTTON IS  
RELEASED, THE DELAY COUNTER IS RESET.

BSEIRE

6C9C	A560	LDA	JSBATA	
6C9E	2920	AND	#\$20	
6CA0	D007	BNE	BSFI10	: BUTTON RELEASED
6CA2	A581	LDA	BSECNT	: GET BUCK SHIP FIRE COUNT
6CA4	E008	BEQ	BSFI20	: OK TO FIRE
6CA6	C681	DEC	BSECNT	: DEC COUNTER AND EXIT
6CA8	60	RTS		
6CA9		BSFI10		
6CA9	A900	LDA	#0	: RESET COUNTER WHEN BUTTON RELEASED
6CAB	8581	STA	BSECNT	
6CAD	60	RTS		

CREATE BUCK SHOT AS TYPE 5, AT (BUCKX AND \$F8)+8, BUCKY-3) USING PATH0.

BSFI20

6CAE	A205	LDX	#5	: CREATE WITH TYPE 5
6CB0	A000	LDY	#0	: SEARCH FROM TOP (BUCK SHOT GETS TOP 2)
6CB2	20E66C	JSR	CREA10	: CREATE OBJECT DESCRIPTOR
6CB5	D028	BNE	BSFI30	: NO ROOM
6CB7	A59A	LDA	BUCKX	: X=BUCKX+8
6CB9	18	CLC		
6CBA	A908	ADC	#8	
6CBC	990106	STA	OBJTBL+1,Y	
6CBE	A59B	LDA	BUCKY	: Y=BUCKY-3
6CD1	38	SEC		
6CD2	E903	SEC	#3	
6CD4	990206	STA	OBJTBL+2,Y	
6CC7	A9C0	LDA	#\$C0	: SPEED=12
6CC9	990506	STA	OBJTBL+5,Y	
6CCC	A905	LDA	#5	: SIZE=5
6CCE	990306	STA	OBJTBL+3,Y	
6CD1	A900	LDA	#0	: SET PATH POINTER
6CD3	200E6D	JSR	SETPTH	
6CD6	A5AF	LDA	FIRDLY	: RESET FIRE DELAY COUNTER
6CD8	8581	STA	BSFCNT	
6CDA	A907	LDA	#7	
6CDC	20A863	JSR	SNDINI	
6CDF		BSFI30		
6CDE	60	RTS		

```

; ROUTINE CREATES A NEW OBJECT DESCRIPTOR INITIALIZING
; THE TYPE FROM X. RETURNS ACC=FF IF NO ROOM FOUND FOR
; OBJECT. OTHERWISE, ACC=0, Y=INDEX TO OBJECT DESCRIPTOR FOUND.
; TYPE BYTE IN OBJECT DESCRIPTOR IS SET FROM X GIVEN, AND ALL OTHER
; BYTES IN OBJECT DESCRIPTOR ARE SET=0. ENTRY POINT AT CREA10
; ALLOWS CREATING A PROCESS GIVEN THE SEARCH STARTING POINT IN Y.
; CALLS TO CREATE DO NOT LOOK AT TOP 2 OBJECT DESCRIPTOR LOCATIONS
; AS THEY ARE RESERVED FOR BUCK SHOTS.
    
```

```

1111 6CE0      CREATE
1112 6CE0 A010      LDY      #16      ; BEGIN SEARCH SKIPPING TOP 2 IN TABLE
1113 6CE2      CREA5
1114 6CE2 C000      CPY      #0      ; CHECK LIMIT
1115 6CE4 F00D ^6CF3 BEQ      CREA20   ; NO ROOM
1116 6CE6 B90006    LDA      OBJTBL,Y
1117 6CE9 F00B ^6CE6 BEQ      CREA30   ; FOUND SPACE
1118 6CEB 98        TYA
1119 6CEC 18        CLC
1120 6CED 6908      ADC      #8
1121 6CEF A8        TAY
1122 6CE0 4CE26C    JMP      CREA5
1123 6CF3      CREA20
1124 6CF3 A9FF      LDA      #$FF     ; NO ROOM
1125 6CF5 60        RTS
1126 6CF6      CREA30
1127 6CF6 8A        TXA
1128 6CF7 990006    STA      OBJTBL,Y ; SET TYPE
1129 6CFA A207      LDX      #7      ; SET OTHER 7 BYTES IN TABLE=0
1130 6CFC A900      LDA      #0
1131 6CFE C8        INY
1132 6CFE      CREA40
1133 6CFE 990006    STA      OBJTBL,Y
1134 6D02 C8        INY
1135 6D03 CA        DEX
1136 6D04 D0F9 ^6CFE BNE      CREA40
1137 6D06 98        TYA
1138 6D07 38        SEC
1139 6D08 E908      SBC      #8
1140 6D0A A8        TAY
1141 6D0B A900      LDA      #0      ; EVERYTHING SET
1142 6D0D 60        RTS
    
```



ROUTINE SETS PATH BYTES IN PTHPTR TO HEAD OF PATH # IN ACC  
Y=INDEX TO OBJECT DESCRIPTOR

: 0A-0A = Y SAVE

SETPH

6D0E				
6D0E	840A	STY	10	: SAVE Y
6D10	48	PHA		: SAVE PATH #
6D11	98	TYA		: GET POINTER 74 FOR PTHPTR INDEX
6D12	4A	LSR	A	
6D13	4A	LSR	A	
6D14	AA	TAX		: SET INDEX
6D15	68	PLA		: GET BACK PATH#
6D16	0A	ASL	A	: *2 FOR TABLE INDEX
6D17	A8	TAY		: SET INDEX
6D18	B9C875	LDA	PTHBL,Y	
6D1B	9520	STA	PTHPTR,X	
6D1D	B9C975	LDA	PTHBL+1,Y	
6D20	9521	STA	PTHPTR+1,X	
6D22	A40A	LDY	10	: RESTORE Y
6D24	60	BTS		

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 48

SCROLL AND BACK DROP INITIALIZATION J:5200 .A65

: BACK DROP INITIALIZATION

: SETS UP MOUNTAIN RANGE AS CARDS 1-23, SCROLLING PATTERN  
: CARDS' COLORS, MOUNTAIN CARDS' COLORS, AND BLANKS.

: CARD USAGE:

: 00-00 = BLANK  
: 01-18 = MOUNTAINS  
: 19-F0 = BITMAP  
: F1-FA = FUEL SCORE  
: FB-FB = SHIPS LEFT  
: EC-ED = DEB COUNT  
: ED-FF = TIME GRAPH

: CONTROL REGISTERS INITIALIZED AS FOLLOWS:

BDINIT

6D25		LDA	#\$26
6D25	A926	STA	COLOR0
6D27	8D1109	LDA	#\$0F
6D2A	A90F	STA	COLOR1
6D2C	8D1209	LDA	#\$00
6D2E	A900	STA	COLOR2
6D31	8D1309	STA	COLOR3
6D34	8D1409	STA	COLOR4
6D37	8D1509		

6D3A	8500	STA	0
6D3C	A920	LDA	#\$20
6D3E	8501	STA	1

BDIN10

6D40	A000	LDY	#\$0
6D42	98	TYA	

BDIN20

6D43	9100	STA	[0],Y
6D44		DEY	

6D40 A000 LBY #40

6D42 98 TYA

6D43 BDIN20

6D43 9100 STA [0],Y

6D45 88 DEY

6D46 D0EB ^6D43 BNE BDIN20

6D46 E601 INC 1

6D4A A501 LDA 1

6D4C C940 CMP #40

6D4E D0E0 ^6D40 BNE BDIN10

6D50 A205 LDX #5

6D52 BD766D QAZL LDA SCDAT,X

6D55 9D2920 STA \$2029,X

6D58 CA DEX

6D59 10F7 ^6D52 BPL QAZL

6D5B A205 LDX #5

6D5D BD7C6D QAZK LDA IIDAT,X

6D60 9D3D20 STA \$203D,X

6D63 CA DEX

6D64 10F7 ^6D5D BPL QAZK

6D66 A5AC LDA SYSTAT

6D68 2902 AND #Z000000010

6D6A D004 ^6D70 BNE SKIPMT

6D6C 20B26B JSR MTNW90

6D6E	60		RTS	
6D70	A940	SKIPMT	LDA	#X01000000
6D72	8D0ED4		STA	NMIEN
6D75	60		RTS	
6D76	514B4E504C	SCDAT	DB	81,75,79,80,76,83
6D7C	CDD2CCCE00	TIDAT	DB	77 OR \$C0,82 OR \$C0,76 OR \$C0,78 OR \$C0,0,83 OR \$C0
6D82	A2C0	SETDLI	LDX	#%11000000
6D84	A900		LDA	#0
6D86	8DB600		STA	SCRIPTR
6D89	A240		LDA	#%01000000
6D8B	8D0ED4		STA	NMIEN
6D8E	AD0BD4	SDLQP	LDA	VCOUNT
6D91	C903		CMP	#3
6D93	D0E9 6D8E		BNE	SDLQP
6D95	8E0ED4		STX	NMIEN
6D98	60		RTS	
6D99	A5AE	SETCOL	LDA	PLYLVL
6D9B	4A		LSR	A
6D9C	4A		LSR	A
6D9D	AA		TAX	
6D9E	BDDC6D	UDSCOL	LDA	COLRSL,X
6DA1	8504		STA	4
6DA3	BDE16D		LDA	COLRSH,X
6DA6	8505		STA	5
6DA8	A935		LDA	#LOW SCROLO
6DAA	8500		STA	0
6DAC	A97B		LDA	#HIGH SCROLO
6DAE	8501		STA	1
6DB0	A900		LDA	#LOW RELSCR
6DB2	8502		STA	2
6DB4	A910		LDA	#HIGH RELSCR
6DB6	8503		STA	3
6DB8	A000	XLOOP	LDY	#0
6DBA	B100		LDA	[0],Y
6DBC	A8		TAY	
6DBD	B104		LDA	[4],Y
6DBE	A000		LDY	#0
6DC1	9102		STA	[2],Y
6DC3	E600		INC	0
6DC5	D002 6DC9		BNE	GLZ9
6DC7	E601		INC	1
6DC9	E602	GLZ9	INC	2
6DCB	D002 6DCE		BNE	GLZ10
6DCD	E603		INC	3
6DCE	A501	GLZ10	LDA	1
6DD1	C97D		CMP	#HIGH SCREND
6DD3	D0E3 6DB8		BNE	XLOOP
6DD5	A500		LDA	0
6DD7	C915		CMP	#LOW SCREND
6DD9	D0DD 6DB8		BNE	XLOOP
6DDB	60		RTS	
6DDC	E0E2F4E6F8	COLRSL	DB	LOW (CLEV1-6),LOW (CLEV2-6),LOW (CLFV3-6),LOW (CLEV4-6),LOW (CLEV5-6)
6DE1	6D6D6D6D6D	COLRSH	DB	HIGH (CLEV1-6),HIGH (CLEV2-6),HIGH (CLEV3-6),HIGH (CLEV4-6),HIGH (CLEV5-6)
6DE6	9890	CLEV1	DB	\$98,\$90
6DE8	1A74	CLEV2	DB	\$1A,\$74
6DEA	1DD4	CLEV3	DB	\$1D,\$D4
6DEC	0604	CLEV4	DB	\$06,\$04

6DEF 0804 CLEV5 DB \$08,\$04

; EACH OBJECT IS HANDLED SEPARATELY ACCORDING TO TYPE.  
 ; THE PC TYPE IS USED TO INDEX INTO A TABLE TO LOCATE THE  
 ; PROPER DRIVER. PC TYPES START AT 1. IF 7=9, THEN OBJECT  
 ; IS NOT TO BE DISPLAYED AND A FLAG IS SET.

; IN ALL CASES, XD=X, YD=Y

; ENTER:

; Y=INDEX INTO OBJTBL FOR OBJECT DESCRIPTOR

; EXIT:

; 00-01 = USED

; A=0 IS DISPLAYABLE

; A=FF IS NOT DISPLAYABLE

6DF0	PCXYAC		
6DE0 B90306	LDA	OBJTBL+3,Y	; SEE IF DISPLAYABLE
6DE3 C909	CMP	#9	
6DE5 9003 6DEA	BCC	PCXY10	
6DE7	PCXY05		
6DF7 A9FF	LDA	#\$FF	; SET NON-DISPLAYABLE FLAG
6DF9 60	RTS		
6DEA	PCXY10		
6DEA B90206	LDA	OBJTBL+2,Y	
6DED C990	CMP	#144	
6DEE B0E6 6DE7	BCC	PCXY05	; BELOW SCREEN - FORGET IT
6E01 B90006	LDA	OBJTBL+0,Y	
6E04 AA	TAX		
6E05 BDD079	LDA	MCODEL-1,X	
6E08 8500	STA	0	
6E0A BDD079	LDA	MCODEH-1,X	
6E0D 8501	STA	1	
6E0F B90306	LDA	OBJTBL+3,Y	
6E12 A8	TAY		
6E13 B100	LDA	[\$00],Y	
6E15 856B	STA	ADDR	
6E17 98	TYA		
6E18 18	CLC		
6E19 690A	ADC	#10	
6E1B A8	TAY		
6E1C B100	LDA	[\$00],Y	
6E1E 856C	STA	ADDR+1	
6E20 A40A	LDY	10	
6E22 B90006	LDA	OBJTBL+0,Y	
6E25 AA	TAX		
6E26 BDB27A	LDA	MCODEL-1,X	
6E29 8500	STA	0	
6E2B BDBB7A	LDA	MCODEH-1,X	
6E2E 8501	STA	1	
6E30 B90306	LDA	OBJTBL+3,Y	
6E33 A8	TAY		
6E34 B100	LDA	[\$00],Y	
6E36 856D	STA	ADDRM	
6E38 98	TYA		
6E39 18	CLC		



6E3A	690A	ADC	#10
6E3C	A8	TAY	
6E3D	B100	LDA	[#001,Y
6E3E	856E	STA	ADDRM+1
6E41	A40A	LDY	10
6E43	B90106	LDA	OBJTBL+1,Y
6E46	851D	STA	\$1D
6E48	B90206	LDA	OBJTBL+2,Y
6E4B	851E	STA	\$1E
6E4D	A900	LDA	#\$0
6E4F	60	RTS	

FOR NORMAL SCALED OBJECTS, PC TYPE IS USED TO INDEX  
 INTO A TABLE POINTING TO THE BASE OF A 36 BYTE TABLE  
 CONTAINING ALL THE PC NUMBERS FOR ALL 4 DIFFERENT XL VALUES  
 AND ALL 9 POSSIBLE SIZES OF PICTURES. 7 IS THEN USED TO GET  
 THE ONE OF FOUR PICTURES PER SIZE. THIS CREATES THE  
 SMOOTH SCROLLING.

ENTER:

Y=INDEX INTO OBJECT DESCRIPTOR  
 X=OBJECT TYPE INDEX (PC TYPE\*2).

6E50	PCSCAL		
6E50 98	TYA		
6E51 48	PHA		
6E52 B90006	LDA	OBJTBL+0,Y	
6E55 AA	IAX		
6E56 BDD079	LDA	DCODEL-1,X	
6E59 8500	STA	0	
6E5B BDD979	LDA	DCODEH-1,X	
6E5E 8501	STA	1	
6E60 B90306	LDA	OBJTBL+3,Y	
6E63 A8	IAY		
6E64 B100	LDA	[0],Y	
6E66 856B	STA	ADDR	
6E68 98	TYA		
6E69 18	CLC		
6E6A 690A	ADC	##A	
6E6C A8	IAY		
6E6D B100	LDA	[0],Y	
6E6F 856C	STA	ADDR+1	
6E71 A009	LDY	##2	:TBL FOR PIX SIZE
6E73 B16B	LDA	[ADDR],Y	
6E75 850B	STA	##B	
6E77 C8	INY		
6E78 B16B	LDA	[ADDR],Y	
6E7A 850C	STA	##C	
6E7C 68	PLA		
6E7D A8	IAY		
6E7E 60	RTS		

```

; ROUTINE MOVES ACTIVE OBJECTS ABOUT DISPLAY AREA.
; OBJECT DATA BASE CONTAINS 8 BYTES PER OBJECT
; ORGANIZED AS FOLLOWS:
;
; 0 = PICTURE CODE (PC)
; 1 = X LOCATION (0-255)
; 2 = Y LOCATION (0-255)
; 3 = Z (SIZE 0-2)
; 4 = COUNT
; 5 = SPEED
; 6 = REPEAT COUNT
; 7 = STATUS
;     7 = MOVED ALREADY
;     6 =
;     5 =
;     4 = MOTHER ZORBA ATTACKING/BUCK SHOT ON REBOUND FROM ZORBA
;     3 = LEFT POLE FLAG
;     2 =
;     1 =
;     0 =
;
; MOVES FURTHEST OBJECTS FIRST, FOLLOWED BY CLOSER OBJECTS.
;
; PC REPRESENTS THE OBJECT TYPE (EG. ROLE, SAUCER, ETC.) AND
; IS TRANSLATED INTO THE PROPER PICTURE TO BE USED BY PAINT
; AND ERASE. (X,Y,Z) IS TRANSLATED TO THE DISPLAY EQUIVALENT
; FORM OF (XD,YD) AND THEN PASSED ON TO PAINT/ERASE.
;
; ENTER:
;
; EXIT:
;
; 00-07 = USED BY PAINT/ERASE
; 08-08 = TEMP FOR MOVE
; 09-09 = INCOMPLETE FOUND FLAG
; 0A-0A = OBJECT TABLE INDEX TO USE
; 10-1B = USED BY PAINT/ERASE
; 1C-1C = PC NUMBER TO DISPLAY
; 1D-1D = XD
; 1E-1E = YD
; 1F-1F = FG COLOR

```

```

46 6E7F      MQVE
47 6E7F A000      LDY      #0          ; SET UP INDEXES
48 6E81 A220      LDX      #32
49 6E83 A9FF      LDA      #$FF
50 6E85 8508      STA      8
51 6E87 A900      LDA      #0
52 6E89 8509      STA      9
53 6E8B      MQVE20
54 6E8B B90006     LDA      OBJTBL,Y ; GET PIC CODE TYPE
55 6E8E F017 ^6EA7 BEQ      MOVE30    ; IF ZERO THEN GET NEXT ENTRY
56 6E90 B90706     LDA      OBJTBL+7,Y ; CHECK HI BIT OF STATUS REGISTER
57 6E93 3012 ^6EA7 BMI      MOVE30    ; DONE ALREADY
6E95 A9FF      LDA      #$FF          ; FOUND ONE INCOMPLETE
6E97 8509      STA      9              ; SET FLAG

```

6E95 A9FF  
6E97 8509

LDA  
STA #FF  
9

FOUND ONE INCOMPLETE  
SET FLAG

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 55  
ROUTINE FOR MOVING OBJECTS ABOUT DISPLAY J:5200 .A65

```
6E99 A508 LDA 8 ; GET LAST SMALLEST Y
6E9B D90206 CMP OBJTBL+2,Y
6E9E 9007 ^6EA7 BCC MOVE30 ; THIS ONE IS CLOSER THAN LAST
6EAB B90206 LDA OBJTBL+2,Y ; THIS IS FURTHEST SO FAR
6EA3 8508 STA 8
6EA5 840A STY 10 ; USE THIS ONE
6EA7 MOVE30
6EA7 98 TYA ; CHECK NEXT
6EA8 18 CLC
6EA9 6908 ADC #8
6EAB A8 TAY
6EAC CA DEX
6EAD D0DC ^6E8B BNE MOVE20 ; DO IT 16 TIMES
6EAF A509 LDA 9 ; SEE IF FOUND ANY INCOMPLETE
6EB1 D020 ^6ED3 BNE MOVE40 ; YES - PROCESS THIS ONE
6EB3 A000 LDY #0 ; CLEAR ALL MOVED BITS FOR NEXT TIME
6EB5 A220 LDX #32
6EB7 MOVE35
6EB7 A97F LDA #7F
6EB9 390706 AND OBJTBL+7,Y
6EBB 990706 STA OBJTBL+7,Y
6EBF 98 TYA
6EC0 18 CLC
6EC1 6908 ADC #8
6EC3 A8 TAY
6EC4 CA DEX
6EC5 D0F0 ^6EB7 BNE MOVE35
6EC7 A57B LDA LVLFLG
6EC9 F007 ^6ED2 BEQ NOGAIN
6ECB A900 LDA #0
6ECD 857B STA LVLFLG
6EEF 400064 JMP LVL0VR
6ED2 60 NOGAIN RTS
6ED3 MOVE40
6ED3 A40A LDY 10 ; GET INDEX INTO OBJTBL
6ED5 B90706 LDA OBJTBL+7,Y ; SET AS OBJECT COMPLETED
6ED8 0980 ORA #80
6EDA 990706 STA OBJTBL+7,Y
6EDD 20F06D JSR PCXYAC ; SET UP XD,YD,PC AND COLOR
6EE0 D003 ^6EE5 BNE MOVE50 ; NOT TO BE ERASED
6EE2 201868 JSR ERASE ; ERASE CURRENT
6EE5 MOVE50
6EE5 A40A LDY 10 ; RESET INDEX
6EE7 20036F JSR MOVPOS ; MOVE OBJECT POSITION
6EEA D00D ^6EF9 BNE MOVE70
6EEC A40A LDY 10 ; RESET INDEX
6EEE 20F06D JSR PCXYAC ; SET UP XD,YD,PC AND COLOR
6EF1 D003 ^6EF6 BNE MOVE60 ; NOT TO BE DRAWN
6EF3 205A67 JSR PAINT ; DRAW IT
6EF6 MOVE60
6EF6 4C7F6E JMP MOVE ; GO DO NEXT ONE
6EF9 MOVE70
6EF9 A40A LDY 10 ; RESET INDEX
6EEB A900 LDA #0 ; SET PC=0 TO CANCEL OBJECT
6EFD 990006 STA OBJTBL,Y
6F00 4C7F6E JMP MOVE ; NEXT...
```

MOVE OBJECT SPEED/16 PIXELS IN THE DIRECTION OF THE  
 DIRECTION BYTE. THE AMOUNT TO MOVE IN ANY PASS IS CALCULATED  
 AS FOLLOWS:

DISTANCE=((OLD COUNT+SPEED)/16)+((SCRSPD\*(9-Z))/256)-(OLD COUNT/16)

ENTER:

Y INDEX SET TO OBJECT DESCRIPTOR

EXIT:

00-04 = USED

03-03 = DIRECTION BYTE SAVE

04-04 = X INDEX SAVE (Y/4)

6E03	MOVPO3				
6E03 98	TYA			SET UP X FOR PTHPTR	
6E04 4A	LSR	A		=OBJTBL INDEX/4	
6E05 4A	LSR	A			
6E06 AA	TAX			GET TO X	
6E07 8604	STX	4		SAVE INDEX	
6E09	MOVPO3				
6E09 B90406	LDA	OBJTBL+6,Y		GET RPTCNT	
6E0C D006 ^6F14	BNE	MOVPO5		OK TO MOVE	
6E0E 200870	JSR	NXTBPT		SET EDINTERS	
6E11 F0F6 ^6F09	BEQ	MOVPO3		NOW MOVE IT	
6E13 60	RTS			GONE	
	GET DISTANCE TO MOVE				
6F14	MOVPO5				
6F14 B90006	LDA	OBJTBL,Y		IF BUCK SHOT, SCRSPD OFFSET=0	
6F17 C905	CMP	#5			
6F19 D007 ^6F22	BNE	MOVPO6			
6F1B A900	LDA	#0		SET SCRSPD OFFSET=0	
6F1D 8502	STA	2			
6F1F 4C316F	JMP	MOVPO7			
6F22	MOVPO6				
6F22 A909	LDA	#2		CALCULATE SCRSPD*(9-Z)/256 FOR Y ADJUST	
6F24 38	SEC				
6F25 F90306	SBC	OBJTBL+3,Y			
6F28 A699	LDX	SCRSPD			
6F2A 20C273	JSR	MULT			
6F2D 8502	STA	2		A=256	
6F2F 0602	ASL	2			
6F31	MOVPO7				
6F31 A604	LDX	4		RESTORE X	
6F33 B90406	LDA	OBJTBL+4,Y		GET OLD COUNT	
6F36 8501	STA	1		SAVE OLD COUNT	
6F38 4A	LSR	A		GET OLD COUNT/16	
6F39 4A	LSR	A			
6F3A 4A	LSR	A			
6F3B 4A	LSR	A			
6F3C 8500	STA	0		SET OLD COUNT/INT	
6F3E A501	LDA	1		GET BACK OLD COUNT	
6F40 18	CLC			GET (OLD COUNT+SPEED)/16	
6F41 790506	ADC	OBJTBL+5,Y			



SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 57  
 MOVE POSITION OF OBJECT J:5200 .A65

```

6F44 920406 STA OBJTBL+4,Y ; UPDATE OLD COUNT
6F47 6A ROR ; KEEP CARRY DURING /16
6F48 4A LSR
6F49 4A LSR
6F4A 4A LSR
6F4B 18 CLC ; ADD TO SCRSED ADJUSTMENT
6F4C 6502 ADC
6F4E 38 SEC ; ALWAYS < 256
6F4F E500 SBC 0 ; SUBTRACT OUT OLD COUNT/16
6F51 MOVE10
6F51 8500 STA 0 ; SAVE
6F53 A604 LDX 4 ; GET INDEX INTO RTHPTR
6F55 A120 LDA RTHPTR,X1 ; GET DIRECTION BYTE
6F57 8503 STA ; TO 3
6F59 A500 LDA
6F5B E02B ^6E88 BEQ MOVP40 ; NOWHERE TO MOVE
6F5D 38 SEC
6F5E E90606 SBC OBJTBL+6,Y ; -RPTCNT
6F61 B00E ^6F72 BCS MOVP20 ; DISTANCE >= RPTCNT
;
; DISTANCE TO MOVE LESS THAN RPTCNT VALUE. SIMPLY REPEAT
; DIR MOVE (DISTANCE) TIMES AND REDUCE RPTCNT BY DISTANCE.
;
6F63 49FF EOR #$FF ; NEW RPTCNT=RPTCNT-DISTANCE
6F65 6901 ADC #1 ; C=0
6F67 990606 STA OBJTBL+6,Y ; UPDATE REPEAT COUNT
6F6A 20D46F JSR MVXY ; MOVE IN XY BY DISTANCE IN 0
6F6D D059 ^6EC8 BNE MOVP100 ; GONE
6F6E 4C886F JMP MOVP40 ; DO X MINSPE DEESET AND EXIT
;
; DISTANCE TO MOVE GREATER THAN (OR EQUAL TO) RPTCNT. FINISH
; OUT REMAINING RPTCNT MOVES AND GET NEW REPEAT COUNT/DIR BYTE
; AND FINISH OUT DISTANCE COUNT
;
6F72 MOVP20
6F72 8502 STA 2 ; SAVE DISTANCE REMAINING
6F74 B90606 LDA OBJTBL+6,Y ; GET CURRENT RPTCNT
6F77 8500 STA 0 ; SET AS DISTANCE TO MOVE HERE
6F79 20D46F JSR MVXY ; FINISH OUT THIS RPTCNT
6F7C D04A ^6EC8 BNE MOVP100 ; GONE
6F7E 200870 JSR NXTRPT ; GET NEXT REPEAT MOVE & REPEAT COUNT
6F81 D045 ^6FC8 BNE MOVP100 ; GONE
6F83 A502 LDA 2 ; GET REMAINING FOR THIS MOVE
6F85 4C516F JMP MOVP10
;
; DO X ADJUSTMENT BY MINSPE.
; ADJUSTMENT ONLY FOR HOPPERS, POLES AND SALICERS
;
6F88 MOVP40
6F88 B90006 LDA OBJTBL,Y ; SEE WHAT THIS IS
6F8B C904 CMP #4
6F8D R036 ^6FC5 BCS MOVP90 ; NO MINSPE ADJUSTMENT
6F8F BE0304 LDX OBJTBL+3,Y ; GET SIZE TO X
6F92 E009 CPX #9 ; NO X ADJUSTMENT FOR SIZE 9
6F94 E02E ^6FC5 BEQ MOVP90
6F96 BDCB6F LDA XSHIFT,X ; GET SHIFT COUNT FROM TABLE
6F99 8500 STA 0
6F9B A596 LDA MTNSPD
6F9D 08 PHP ; DIVIDE MTNSPD BY 8*101
6F9E 1005 ^6FA5 BPL MOVP50 ; SAVE SIGN OF MTNSPD
; FORCE POSITIVE #

```

6FA0	49FF	EOR	#\$FF	
6FA2	18	CLC		
6FA3	6901	ADC	#1	
6FA5		MOVP50		
6FA5	4A	LSR	A	; DIVIDE BY 2
6FA6	C600	DEC	0	
6FA8	00FB ^6FA5	BNE	MOVP50	
6FAA		MOVP60		
6FAA	28	PLP		; GET SIGN OF MTNSPD
6FAB	1005 ^6FB2	BPL	MOVP70	
6EAD	49FF	EOR	#\$FF	; MAKE NEG AGAIN
6EAF	18	CLC		
6EB0	6901	ADC	#1	
6EB2		MOVP70		
6EB2	08	PHP		; SAVE SIGN OF MTNSPD
6EB3	18	CLC		; ADD TO X POSITION
6EB4	720106	ADC	OBJTBL+1,Y	
6EB7	720106	STA	OBJTBL+1,Y	
6EBA	B006 ^6FC2	BCC	MOVP80	; CHECK BOUNDARIES
6EBC	28	PLP		; GET BACK SIGN
6EBD	3009 ^6FC8	BMI	MOV100	; OUT OF AREA
6EBE	40C5AF	JMP	MOVP90	; STILL OK
6FC2		MOVP80		
6FC2	28	PLP		; GET BACK MTNSPD SIGN
6FC3	1003 ^6FC8	BPL	MOV100	; OUT OF AREA
6FC5		MOVP90		
6FC5	A900	LDA	#0	; SET STILL IN BOUNDS
6FC7	60	RTS		
6FC8		MOV100		
6FC8	A9FF	LDA	#\$FF	; OUT OF DISPLAY - TO BE CANCELED
6FCA	60	RTS		
6FCB		XSHIFT		
6FCB	0203030404	DB	2,3,3,4,4,4,5,5,5	

## XY MOVE ROUTINE

J:5200 .A65

; MOVE XY LOCATIONS TO 1 UNITS GIVEN DIR BYTE IN B

```

6FD4      MVXY
6FD4 20E36F      JSR      MVXY30      ; MOVE X
6FD7 D009 ^6FE2  BNE      MVXY20      ; GONE
6FD9 C8          INY          ; SET UP TO MOVE Y
6FDA 20E36F      JSR      MVXY30
6FDD D019 ^6FF8  BNE      MVXY40      ; GONE
6FDE 88          DEY          ; RESET Y
6FE0      MVXY10
6FE0 A900      LDA      #0          ; STILL OK
6FE2      MVXY20
6FE2 60          RTS
6FE3      MVXY30
6FE3 2603      ROL      3          ; SEE IF TO MOVE THIS ONE
6FE5 90E9 ^6FE0  BCC      MVXY10      ; NO - STILL OK
6FE7 A910      LDA      #16        ; POS/NEG?
6FE9 2503      AND      3
6FEB F00E ^6FFB  BEQ      MVXY50      ; NEG
6FED B90106     LDA      OBJTBL+1,Y  ; POSITIVE MOVE
6FE0 18        CLC          ; ADD IN DISTANCE
6FE1 6500      ADC      0
6FE3 990106     STA      OBJTBL+1,Y
6FE6 90E8 ^6FE0  BCC      MVXY10      ; OK
6FF8      MVXY40
6FF8 A9FF      LDA      #$FF        ; GONE
6FFA 60        RTS
6FEB      MVXY50
6FEB B90106     LDA      OBJTBL+1,Y  ; SUBTRACT OUT DISTANCE
6FEF E500      SBC      0          ; C=1
7000 990106     STA      OBJTBL+1,Y
7003 B0DB ^6FE0  BCS      MVXY10      ; OK
7005 4CE86F     JMP      MVXY40      ; GONE

```

```

; GETS NEXT REPEAT COUNT HANDLING ANY SPECIAL CONFIGURATION BYTES
;
; MOVE ROUTINE MOVES AN OBJECT ACCORDING TO THE DIRECTION
; AND SPEED. SPEED REPRESENTS THE NUMBER OF SIXTEENTHS OF
; A PIXEL THAT THE OBJECT IS TO MOVE IN THE DIRECTION SPECIFIED
; BY THE DIRECTION BYTE IN THE PATH TABLE. THE PATH TABLE IS POINTED TO
; BY A 32 BYTE TABLE (DIRTBP) IN 0-PAGE (2 BYTES/OBJECT) USED FOR
; INDEXING INTO THE PATH TABLE AND THE INDEX VALUE NORMALLY
; POINTS TO THE DIRECTION BYTE. THE ORDER OF THE PATH TABLE
; IS SIMPLY A REPEAT COUNT BYTE (0-127) FOLLOWED BY THE DIRECTION
; OF TRAVEL. WHEN A NEW DIRECTION BYTE IS READ, THE SPEED AND Z
; VALUES ARE UPDATED. AFTER THAT, ONLY THE X AND Y VALUES ARE
; CHANGED. THE DIRECTION BYTE DEFINES THE DIRECTION OF TRAVEL
; AS FOLLOWS:
;
; BIT 0 = +/-SPEED      1=+, 0=-
; BIT 1 = +/-Z (SIZE)   1=+, 0=-
; BIT 2 = +/-Y          1=+, 0=-
; BIT 3 = +/-X          1=+, 0=-
; BIT 4 = SPEED CHANGE  1=CHANGE, 0=NO CHANGE
; BIT 5 = Z CHANGE      1=CHANGE, 0=NO CHANGE
; BIT 6 = Y CHANGE      1=CHANGE, 0=NO CHANGE
; BIT 7 = X CHANGE      1=CHANGE, 0=NO CHANGE
;
; IF COUNT BIT 7=1, THEN IT IS TO BE INTERPRETED AS A SPECIAL
; CONFIGURATION BYTE FOLLOWED BY ANY ARGUMENTS REQUIRED
; OF THE CONFIGURATION COMMAND. THE CONFIGURATION COMMANDS
; ARE AS FOLLOWS:
;
; 80 = JUMP TO arg. IN PATH TABLE (2 BYTES FOR NEW PTHPTR VALUE)
; 81 = LOAD PC TYPE IMMEDIATE W/ arg.
; 82 = CANCEL ALWAYS
; 83 = CALL SUBROUTINE PATH (NEXT 2 BYTES ARE ADDRESS OF SUBROUTINE)
; 84 = RETURN TO ORIGINAL PATH
; 85 = LOAD SPEED DIRECT
; 86 = CHECK BUCK BETWEEN POLES
; 87 = MOTHER ZORBA TURNS AROUND
;
; ENTER:
;
; Y=OBJTBL INDEX FOR OBJECT TO MOVE
;
; EXIT:
;
; Q0-Q2 = USED
; X,Y,Z,S UPDATED
;
; PTHPTR POINTS TO NEXT DIRECTION BYTE. REPEAT COUNT UPDATED
; IN OBJECT DESCRIPTOR TABLE. Z AND SPEED ARE UPDATED AS REQUIRED.
;
; IF A=0, STILL IN DISPLAY AREA.
; IF A=FF, MOVED OUT OF DISPLAY AREA.

```

```

7008      NXTRPT
7008 20BB73 JSR      INDRPT      ; INC POINTER TO NEXT REPEAT COUNT
700E      NXTR10
700B A120   LDA      [PTHPTR,X]  ; GET NEW REPEAT COUNT
700D 3003 ^7012 BMI    NXTR20    ; SPECIAL CONFIGURATION BYTE

```

700B A120 LDA [PTHPTR,X] ; GET NEW REPEAT COUNT  
 700D 3003 ^7012 BMI NXTR20 ; SPECIAL CONFIGURATION BYTE

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 61  
 J:5200 .A65

NEXT REPEAT COUNT GETTER

```

700E 4C7E70 JMP NXTR70 ; GO TO 7, SPEED AND EXIT
7012 NXTR20 ; SET UP FOR TABLE JUMP
7012 297F AND #7F
7014 C908 CMP #8
7016 B0FA ^7012 BCS NXTR20 ; SEE IF ANYTHING INVALID
7018 0A ASL A
7019 A8 TAY
701A B92970 LDA CONFGT,Y
701D 8500 STA 0
701E B92A70 LDA CONFGT+1,Y
7022 8501 STA 1
7024 A40A LDY 10 ; RESTORE Y
7026 6C0000 JMP 101
7029 CONFGT
7029 4E70 DW NXTR40 ; JUMP
702B 4570 DW NXTR30 ; LOAD PC IMMEDIATE
702D 3970 DW NXTR21 ; CANCEL
702E 5E70 DW NXTR50 ; CALL
7031 6B70 DW NXTR60 ; RETURN
7033 3C70 DW NXTR25 ; SET SPEED
7035 D870 DW NXTR140 ; CHECK BUCK BETWEEN POLES & DEC UFOCNT
7037 2571 DW NXTR190 ; MOTHER ZORBA TURN AROUND
7039 NXTR21
7039 A9EF LDA #8EF ; CANCEL THIS ONE
703B 60 RTS
;
; SET SPEED
703C NXTR25
703C 20B673 JSR GETNXT ; SET SPEED
703E 990506 STA OBJTBL+5,Y
7042 4C0B70 JMP NXTR10
;
; LOAD PC TYPE IMMEDIATE W/ NEXT DATA BYTE
7045 NXTR30
7045 20B673 JSR GETNXT ; GET ARGUMENT
7048 990006 STA OBJTBL,Y ; SET TYPE
704B 4C0B70 JMP NXTR10 ; NEXT
;
; JUMP TO NEW LOCATION FOR PTHPTR
704E NXTR40
704E 20B673 JSR GETNXT ; POINT TO ARGUMENT
7051 8500 STA 0 ; SET NEW PTHPTR LOW BYTE
7053 A120 LDA [PTHPTR,X] ; GET NEW PTHPTR HI BYTE
7055 9521 STA PTHPTR+1,X ; SET NEW PTHPTR HIGH BYTE
7057 A500 LDA 0 ; SET NEW PTHPTR LOW BYTE
7059 9520 STA PTHPTR,X
705B 4C0B70 JMP NXTR10 ; ALL SET FOR NEXT
;
; CALL IN PATH TABLE - SAVES CURRENT PATH POINTER VALUE
; AND SETS NEW PATH POINTER FROM NEXT TO ARG'S.
705E NXTR50
705E B520 LDA PTHPTR,X ; SAVE ADDRESS
7060 9D0008 STA RTNADD,X
7063 B521 LDA PTHPTR+1,X
7065 9D0108 STA RTNADD+1,X
7068 4C4E70 JMP NXTR40 ; JUMP TO NEW ADDRESS

```



; RETURN IN PATH TABLE - RESTORES ADDRESS, POINTS TO NEXT ENTRY  
; AND CONTINUES

```

704B      NXTR60
704B BD0008      LDA      RTNADD,X      ; RESTORE ADDRESS
704E 9520          STA      PTHPTR,X
7070 BD0108      LDA      RTNADD+1,X
7073 9521          STA      PTHPTR+1,X
7075 20BB73      JSR      INDRPT      ; SKIP OVER CALL INSTRUCTION
7078 20BB73      JSR      INDRPT      ; SKIP OVER ARG1
707B 4C0870      JMP      NXTRPT      ; EXECUTE NEXT COMMAND SKIPPING ARG2

```

; MOVE Z, SPEED AND EXIT

```

707E      NXTR70
707E 990606      STA      OBJTBL+6,Y      ; SET REPEAT COUNT
7081 20BB73      JSR      INDRPT      ; INC POINTER TO NEXT DIRECTION BYTE
7084 A120          LDA      [PTHPTR,X]      ; GET DIR BYTE TO 1
7086 8501          STA      1
7088 2920          AND      #20      ; SEE IF TO MOVE IN Z
708A F021 ^70AD   BEQ      NXTR90      ; NO
708C A902          LDA      #2      ; SEE IF POS/NEG
708E 2501          AND      1
7090 F010 ^70A2   BEQ      NXTR80      ; NEG
7092 B90306      LDA      OBJTBL+3,Y      ; POS - INC Z
7095 18            CLC
7096 6901          ADC      #1
7098 C90A          CMP      #10      ; 9 IS MAX
709A B011 ^70AD   BCS      NXTR90      ; TOO HIGH - LEAVE IT AND DO SPEED
709C 990306      STA      OBJTBL+3,Y
709F 4CAD70      JMP      NXTR90
70A2      NXTR80
70A2 B90306      LDA      OBJTBL+3,Y      ; DEC Z
70A5 38            SEC
70A6 E901          SRC      #1
70A8 3003 ^70AD   BMI      NXTR90      ; LOWER LIMIT EXCEEDED - LEAVE IT
70AA 990306      STA      OBJTBL+3,Y

```

; NOW DO SPEED

```

70AD      NXTR90
70AD A910          LDA      #10      ; SEE IF TO MOVE IN SPEED
70AF 2501          AND      1
70B1 F013 ^70C6   BEQ      NXT110      ; NO - DONE
70B3 A201          LDA      #1      ; SEE IF POS/NEG
70B5 2501          AND      1
70B7 F010 ^70C9   BEQ      NXT120      ; NEG
70B9 B90506      LDA      OBJTBL+5,Y      ; ADD TO SPEED
70BC 18            CLC
70BD 6908          ADC      #8
70BF 90D2 ^70C3   BEQ      NXT100
70C1 A2FF          LDA      #FF      ; MAX AT FF
70C3      NXT100
70C3 990506      STA      OBJTBL+5,Y
70C6      NXT110
70C6 A900          LDA      #0      ; OK
70C8 60            RTS      ; DONE
70C9      NXT120
70C9 B90506      LDA      OBJTBL+5,Y      ; NEG

```

70C6	A900	LDA	#0	: OK
70C8	60	RTS		: DONE
70C9				
70C9	B90506	LDA	OBJTBL+9,Y	: NEG

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 63  
NEXT REPEAT COUNT GETTER J:5200 .A65

70CC	38	SEC		
70CD	E908	SBC	#8	: SUBTRACT FROM SPEED
70CF	F002 ^70D3	BEO	NXT130	: ZERO NO GOOD
70D1	B0F0 ^70C3	BES	NXT100	
70D3			NXT130	
70D3	A901	LDA	#1	: MINIMUM AT 1
70D5	4CC370	JMP	NXT100	
				: GOT CONFIGURATION COMMAND TO CHECK BUCK BETWEEN POLES.
				: A CHECK IS MADE TO SEE IF ANY OTHER POLE HAS BEEN FOUND YET.
				: IF NOT, THIS POLE'S X IS SAVED AND THE PTHPTR IS INC'D. IF NOT,
				: A CHECK IS MADE TO SEE IF THIS BUCK IS BETWEEN THIS POLE
				: AND THE LAST POLE CHECKED. IF SO, A POINT BONUS IS AWARDED.
70D8			NXT140	
70D8	A58D	LDA	POLEND	: SEE IF THIS IS FIRST POLE CANCELED
70DA	D00A ^70E6	BNE	NXT150	: NO - SEE IF BUCK BETWEEN THEM
70DC	B90106	LDA	OBJTBL+1,Y	: SET POLEND TO CONTAIN THIS POLES'S X
70DE	0901	ORA	#1	: FORCE NON-ZERO - ERROR NEGLIGABLE
70E1	858D	STA	POLEND	
70E3	4C0870	JMP	NXT150	
70E6			NXT150	
70E6	A59A	LDA	BUCKX	: COMPARE TO SEE IF BUCK BETWEEN
70E8	18	CLC		: NORMALIZE TO FIND CENTER OF BUCK
70E9	6908	ADC	#8	
70EB	D90106	CMP	OBJTBL+1,Y	: BUCKX-THIS POLE X
70EE	9007 ^70F7	BCC	NXT160	: BUCKX < THIS POLE X
70F0	C58D	CMP	POLEND	: BUCKX > THIS POLE X
70F2	9007 ^70FB	BCC	NXT170	: BUCKX < OTHER POLE - BETWEEN THEM
70F4	4C1E71	JMP	NXT180	: BUCKX > OLD POLE, TOO - NOT BETWEEN
70F7			NXT160	
70F7	C58D	CMP	POLEND	
70F9	9023 ^711E	BCC	NXT180	: BUCKX < OLD POLE, TOO - NOT BETWEEN
70FB			NXT170	
70FB	A900	LDA	#0	: BETWEEN POLES - BONUS 500PTS
70FD	8600	STX	0	: SAVE X
70FF	A205	LDX	#5	
7101	208973	JSR	ADDSCR	
7104	A600	LDX	0	: RESTORE X
7106	C6A6	DEC	LIFOCNT	: DEC LIFO COUNT WHEN PASSING THRU POLES
7108	D004 ^710E	BNE	NXT175	
710A	A9FF	LDA	#FF	
710C	857B	STA	LVLFLG	
710E			NXT175	
710E	98	TYA		: SAVE X,Y DESTROYED BY LIFOISP
710F	48	PHA		
7110	8A	TXA		
7111	48	PHA		
7112	208C60	JSR	UFODSR	
7115	A909	LDA	#9	
7117	20A863	JSR	SNDINI	
711A	68	PLA		: RESTORE X,Y
711B	AA	TAX		
711C	68	PLA		
711D	A8	TAY		
711E			NXT180	
711E	A900	LDA	#0	: RESET POLEND
7120	858D	STA	POLEND	
7122	4C0870	JMP	NXT180	: NOW GET NEXT REPEAT COUNT

: MOTHER SHIP TURNS AROUND - GIVE IT ONE OF 2 OF THE SAUCER  
: RETURN PATHS

7125		NXT190		
7125	AD0AE8	LDA	RANDOM	: SEE WHICH ONE TO USE
7128	3005 ^712F	BMI	NXT200	: USE #17
712A	A912	LDA	#18	: USE #18
712C	4C3171	JMP	NXT210	
712E		NXT200		
712F	A911	LDA	#17	
7131		NXT210		
7131	200FAD	JSR	SETPTH	
7134	B90706	LDA	QBUTBL+7,Y	: SET ATTACK BIT
7137	0910	ORA	##10	
7139	990706	STA	QBUTBL+7,Y	
713C	4C0870	JMP	NXTRPT	: NOW MOVE ZORBA BACK

## UTILITIES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 65  
J:5200 .A65

; ROUTINE TO DISPLAY SCORE

; ----&gt; UPDATES SCORE IN DISPLAY FROM CSCORE+3 TO CSCORE WITH

; CSCORE BEING MSB AND CSCORE+3 BEING LSB. THIS IS BACKWARDS FROM NORMAL.

```
713F      SCRUPD
713F A200      LDX      #0
7141 A000      LDY      #0
7143 B00E09     SCRUI0   LDA      CSCORE+1,X
7146 4A        LSR      A
7147 4A        LSR      A
7148 4A        LSR      A
7149 4A        LSR      A
714A 18        CLC
714B 6941      ADC      #41
714D 992F20     STA      $202F,Y
7150 C8        INY
7151 B00E09     LDA      CSCORE+1,X
7154 290F      AND      #0F
7156 18        CLC
7157 6941      ADC      #41
7159 992F20     STA      $202F,Y
715C C8        INY
715D E8        INX
715E E003      CPX      #3
7160 D0E1 ^7143 BNE      SCRUI0
7162 60        RTS
```

; ROUTINE TO DISPLAY TITLE PAGE

; ----> DISPLAYS COPYRIGHT, HIGH SCORE (FROM HISCOR+3 TO HISCOR WITH  
; HISCOR BEING MSB AND HISCOR+3 BEING LSB. BACKWARDS.

```
7163      TITLE
7163 A900      LDA      #0
7165 8D00D4     STA      $D400
7168 A946      LDA      #LOW TLIST
716A 8D0009     STA      $DLSTL
716D A973      LDA      #HIGH TLIST
716E 8D0109     STA      $DLSTH
7172 A9E8      LDA      #$E8 ;@@@
7174 8D0309     STA      $CHBAS
7177 A940      LDA      #40
7179 8D0ED4     STA      $NMIEN
717C A922      LDA      #200100010
717E 8D0209     STA      $DMCTL
7181 A920      LDA      #20
7183 8501      STA      1
7185 A900      ZXCBL    LDA      #0
7187 8500      STA      0
7189 A8        TAY
718A 9100      ZXCLP    STA      $01,Y
718C 88        DEY
718D D0EB ^718A     BNE      ZXCLP
718F E601      INC      1
7191 A501      LDA      1
7193 C940      CMP      #40
7195 D0EE ^7185     BNE      ZXCBL
7197 A200      LDX      #0
```

## UTILITIES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 66  
J:5200 .A65

7199	BD0A72	OPTLOP	LDA	BR1,X
719A	38		SEC	
719B	E920		SBC	##20
719E	9D0020		STA	\$200D,X
71A2	E8		INX	
71A3	F0C8		CPX	#200
71A5	D0F2 ^7199		BNE	OPTLOP
71A7	A200		LDX	##0
71A9	8E1509		STX	COLOR4
71AC	A000		LDY	##0
71AE	BD0A09	HSCOUT	LDA	HISCOR+1,X
71B1	4A		LSR	A
71B2	4A		LSR	A
71B3	4A		LSR	A
71B4	4A		LSR	A
71B5	18		CLC	
71B6	6990		ADC	##90
71B8	990E20		STA	\$200E,Y
71BB	C8		INX	
71BC	BD0A09		LDA	HISCOR+1,X
71BF	290F		AND	##0F
71C1	18		CLC	
71C2	6990		ADC	##90
71C4	990E20		STA	\$200E,Y
71C7	C8		INX	
71C8	E8		INX	
71C9	E003		CPX	#3
71CB	D0E1 ^71AE		BNE	HSCOUT
71CD	A200		LDX	##0
71CF	A000		LDY	##0
71D1	BD0E09	RSCOUT	LDA	CSCORE+1,X
71D4	4A		LSR	A
71D5	4A		LSR	A
71D6	4A		LSR	A
71D7	4A		LSR	A
71D8	18		CLC	
71D9	6990		ADC	##90
71DB	990320		STA	\$2003,Y
71DE	C8		INX	
71DF	BD0E09		LDA	CSCORE+1,X
71E2	290F		AND	##0F
71E4	18		CLC	
71E5	6990		ADC	##90
71E7	990320		STA	\$2003,Y
71EA	C8		INX	
71EB	E8		INX	
71EC	E003		CPX	#3
71EE	D0E1 ^71D1		BNE	RSCOUT
71F0	A968		LDA	##68
71F2	9D1109		STA	COLOR0
71F5	A990		LDA	##90
71F7	8D1209		STA	COLOR1
71FA	A9A8		LDA	##A8
71FC	8D1409		STA	COLOR3
71FE	A968		LDA	##68
7201	8D1309		STA	COLOR2
7204	AD09E8		LDA	KBCODE
7207	85AC		STA	SYSTAT
7209	AD0BD4	RBNW	LDA	VCOUNT
720C	18		CLC	



```
720D 6578      ADC      TIMER
720E 29F0      ANI      #X11110000
7211 0908      ORA      #X00001000
7213 8D0AD4     STA      WSYND
7216 8D18C0     STA      $C018
7219 A578      LDA      TIMER
721B 8D1409     STA      COLOR3
721E AD09E8     LDA      KRCODE
7221 C5AC      CMP      SYSTAT
7223 D006 ^722B BNE      CHKITA
7225 AD10C0     LDA      $C010
7228 D0DE ^7209 BNE      RROW
722A 60        EXITRB  RTS
722B 29DF      CHKITA AND      #X11011111
722D C919      CMP      #X00011001
722E D0D8 ^7209 BNE      RROW
7231 60        RTS
```

## ; PAINT BUCK SHIP/SHADOW

```
7232 A9C5      BSERAS  LDA      #LOW_SHADOW
7234 856B      STA      ADDR
7236 A996      LDA      #HIGH_SHADOW
7238 856C      STA      ADDR+1
U 723A A900     LDA      #LOW_MSHADOW
723C 856D      STA      ADDR+1
U 723E A900     LDA      #HIGH_MSHADOW
7240 856E      STA      ADDR+1
7242 A59A      LDA      BUCKX
7244 851D      STA      $1D
7246 A988      LDA      #131
7248 851F      STA      $1F
724A 201868    JSR      ERASE
724D A680      LDY      LASTBS
724F BDAF72     LDA      BUKSL,X
7252 856B      STA      ADDR
7254 BDB172     LDA      BUKSH,X
7257 856C      STA      ADDR+1
7259 BDB472     LDA      MBUKSL,X
725C 856D      STA      ADDR+1
725E BDB772     LDA      MBUKSH,X
7261 856E      STA      ADDR+1
7263 A59A      LDA      BUCKX
7265 851D      STA      $1D
7267 A59B      LDA      BUCKY
7269 851E      STA      $1E
726B 201868    JSR      ERASE
726E 60        RTS
726F A9C5      BSEAIN  LDA      #LOW_SHADOW
7271 856B      STA      ADDR
7273 A996      LDA      #HIGH_SHADOW
7275 856C      STA      ADDR+1
U 7277 A900     LDA      #LOW_MSHADOW
7279 856D      STA      ADDR+1
U 727B A900     LDA      #HIGH_MSHADOW
727D 856E      STA      ADDR+1
727F A59A      LDA      BUCKX
```

```
7281 851D      STA      $1D
7283 A983      LDA      #121
7285 851E      STA      $1E
7287 205A67    JSR      PAINT
728A A680      LDX      LASTBS
728C B0AE72     LDA      BUKSL,X
728E 856B      STA      ADDR
7291 B0B172     LDA      BUKSH,X
7294 856C      STA      ADDR+1
7296 B0B472     LDA      MBUKSL,X
7299 856D      STA      ADDR+1
729B B0B772     LDA      MBUKSH,X
729E 856E      STA      ADDR+1
72A0 A59A      LDA      BUCKX
72A2 851D      STA      $1D
72A4 A59B      LDA      BUCKY
72A6 851E      STA      $1E
72A8 205A67    JSR      PAINT
72AB 4C9C4C    JMP      BSFIRE
```

; THIS IS A TABLE TO DECODE BUCK ADDRESSES

```
72AE A79DB1     BUKSL     DB      LOW BUCKL,LOW BUCK,LOW BUCKR
72B1 969696     BUKSH     DB      HIGH BUCKL,HIGH BUCK,HIGH BUCKR
```

```
U 72B4 000000   MBUKSL    DB      LOW MBUCKL,LOW MBUCK,LOW MBUCKR
U 72B7 000000   MBUKSH    DB      HIGH MBUCKL,HIGH MBUCK,HIGH MBUCKR
```

```
72BA 53433A     BB1       DB      'SC:'
72BD B0B0B0B0B0  DB      $B0,$B0,$B0,$B0,$B0,$B0
72C3 202048493A DB      'HI:'
72C8 B0B0B0B0B0  DB      $B0,$B0,$B0,$B0,$B0,$B0
72CE 2020202020  DB
72D6 C2D5C3CB     DB      'R'+128,'U'+128,'C'+128,'K'+128
72DA 2020202020  DB
72E2 2020202020  DB
72E9 D2CFC7C5D2   DB      'R'+128,'O'+128,'G'+128,'E'+128,'R'+128,'S'+128
72EF 2020202020  DB
72F6 202020434F   DB      'COPYRIGHT 1983'
730A 2020       DB
730C D3C5C7C1     DB      'S'+128,'E'+128,'G'+128,'A'+128
7310 20454E5445   DB      'ENTERPRISES'
731E 2020202020  DB      'INC.'
7332 5052455353   DB      'PRESS'
7338 1314011214   DB      $13,$14,$01,$12,$14 ; 'START' FLASHING
733D 20544E2042   DB      'TO BEGIN'
```

```
7346 707070     TLIST     DB      $70,$70,$70
7349 4600207070   DB      $46,$00,$20,$70,$70,$70,$70,$07,$07,$70,$70,$50,$06,$20,$06,$20
7359 0670707070   DB      $06,$70,$70,$70,$70,$70,$70,$70,$06
7362 41          DB      $41
7363 4673        DW      TLIST
```

; CHECK FOR NEW SCORE HIGHER THAN OLD SCORE

```
7365 A200      LDX      #0
7367          CHKHS0
```

1	7367	B00009	LDA	CSCORE,X
2	736A	D00909	CMF	HISCOR,X
3	736D	9012 ^7381	BCC	CHKH40
4	736F	D005 ^7376	BNE	CHKH20
5	7371	EA	INX	
6	7372	E004	CPX	#4
7	7374	D0F1 ^7367	BNE	CHKH10
8	7376		CHKH20	
9	7378	A203	LIX	#3
10	7378		CHKH30	
11	7378	B00009	LDA	CSCORE,X
12	737B	9D0909	SIA	HISCOR,X
13	737E	CA	DEX	
14	737F	10F7 ^7378	BPL	CHKH30
15	7381		CHKH40	
16	7381	60	RTS	
17				
18				
19				
20	7382		INSDPT	
21	7382	E662	INC	SNDPTR,X
22	7384	D002 ^7388	BNE	INSD10
23	7386	E663	INC	SNDPTR+1,X
24	7388		INSD10	
25	7388	60	RTS	
26				
27				
28				
29	7389		ADDSCR	
30	7389	18	CLC	
31	738A	FB	SED	
32	738B	6D1009	ADC	CSCORE+3
33	738E	8D1009	STA	CSCORE+3
34	7391	8A	IXA	
35	7392	6D0F09	ADC	CSCORE+2
36	7395	8D0F09	STA	CSCORE+2
37	7398	A900	LDA	#0
38	739A	6D0E09	ADC	CSCORE+1
39	739B	8D0E09	STA	CSCORE+1
40	73A0	A900	LDA	#0
41	73A2	6D0D09	ADC	CSCORE
42	73A5	8D0D09	STA	CSCORE
43	73A8	D8	CLD	
44	73A9	60	RTS	
45				
46				
47				
48				
49	73AA		AD1625	
50	73AA	A505	LDA	5
51	73AC	18	CLC	
52	73AD	6910	ADC	#16
53	73AE	8505	STA	5
54	73B1	9002 ^73B5	BCC	ADD810
55	73B3	E606	INC	6
56	73B5		ADD810	
57	73B5	60	RTS	
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				

INCREMENT SOUND POINTER

ROUTINE TO ADD NUMBER IN X,A (BCD, X IS MSB) TO SCORE

ADD 16 TO REGISTER PAIR [5,6]

ROUTINE FOR GETTING NEXT BYTE FROM PATH TABLE AND POINTING TO BYTE FOLLOWING THAT

```
1 73B4 GETNXT
2 73B4 20BB73 JSR INDRPT
3 73B9 A120 LDA [PTHPTR,X]
4
5 ; ROUTINE FOR INCREMENTING WORD AT PTHPTR+X.
6 ;
7 73BB INDRPT
8 73BB F620 INC PTHPTR,X
9 73BD D002 ^73C1 BNE INDR10
10 73BF F621 INC PTHPTR+1,X
11 73C1 INDR10
12 73C1 60 RTS
13
14 ; MULTIPLY ROUTINE MULTIPLIES TWO 8 BIT NUMBERS IN A,X
15 ; AND CREATES A 16 BIT PRODUCT IN A,X. A CHECK IS
16 ; MADE TO ASSURE THAT MULTIPLICAND IS LESS THAN MULTIPLIER
17 ; SO THAT THE ROUTINE EXECUTES AT MAXIMUM SPEED.
18 ;
19 ; ENTER:
20 ; X=MULTIPLICAND
21 ; A=MULTIPLIER
22 ;
23 ; EXIT:
24 ; 00-02 = USED
25 ; A = RESULT HIGH
26 ; X = RESULT LOW
27 ;
28
29 73C2 MULT
30 73C2 E000 CPX #0 ; TEST IF MULPLICAND=0
31 73C4 F017 ^73DD BEQ MULT30
32 73C6 CA DEX ; DEC MULTIPLICAND TO AVOID
33 73C7 8600 STX 0 ; THE CLC BEFORE ADC 0
34 73C9 4A LSR A ; SHIFT OUT LOW BIT FOR
35 73CA 8501 STA 1 ; THE FIRST ITERATION
36 73CC A900 LDA #0 ; RESULT HI=0
37 73CE A208 LDX #8 ; 8 BITS=8 ITERATIONS
38 73D0 MULT10
39 73D0 9002 ^73D4 BCC MULT20 ; TEST LO BIT OF MULTIPLIER
40 73D2 6500 ADC 0 ; ADD IN TO PRODUCT
41 73D4 MULT20
42 73D4 6A ROR A ; SHIFT RESULT DOWN 1 BIT AND
43 73D5 6601 ROR 1 ; SHIFT OUT NEXT BIT OF MULTIPLIER
44 73D7 8A DEX
45 73D8 D0E6 ^73D0 BNE MULT10
46 73DA A601 LDX 1 ; SET LSB RESULT
47 73DC 60 RTS
48 73DD MULT30
49 73DD 8A TXA ; RESULT=0
50 73DE 60 RTS
51
52 73DF VBIRTN
53 73DF 48 PHA
54 73E0 AD1109 LDA COLOR0 ; COPY RAM COLORS TO HARDWARE REGS.
55 73E3 8D16C0 STA $C016
56 73E6 AD1209 LDA COLOR1
57 73E9 8D17C0 STA $C017
73EC AD1309 LDA COLOR2
73EF 8D18C0 STA $C018
```

```

73E2 AD1409 LDA COLOR3
73E5 8D19C0 STA $E019
73E8 AD1509 LDA COLOR4
73EB 8D1AC0 STA $E01A
73FE A904 LDA ##4
7400 8D1EC0 STA CONSOL ; RESET CONSOL SWITCH
7403 AD0009 LDA SDSLSTL
7406 8D02D4 STA $D402
7409 AD0109 LDA SDSLSTH
740C 8D03D4 STA $D403
740E AD0209 LDA SDMCTL
7412 8D00D4 STA DMACTL
7415 AD00E8 LDA $E800
7418 8D1609 STA PDLO
741B AD01E8 LDA $E801
741E 8D1709 STA PDLI
7421 8D0BE8 STA $E80B
7424 AD0409 LDA CHART
7427 8D01D4 STA CHACTL
742A AD0309 LDA CHBAS
742D 8D09D4 STA CHBASE
7430 E678 INC TIMER
7432 68 PLA
7433 40 RTI
    
```

DO JOYSTICK WORK

READS JOYSTICK AND PLACES DATA IN JSDATA.  
DATA IS AS FOLLOWS:

- : BIT 0-
- : BIT 1-
- : BIT 2- UP
- : BIT 3- DOWN
- : BIT 4- LEFT
- : BIT 5- FIRE
- : BIT 6-
- : BIT 7- RIGHT

```

7434 JOYSTK
7434 A9FE LDA ##FF
7436 8560 STA JSDATA
7438 A59C LDA FULAMT ; SEE IF OUT OF FUEL
743A 059D ORA FULAMT+1
743C 059F ORA FULAMT+2
743E D003 >7443 BNE JOYS10
7440 4C9074 JMP JOYS90
7443 JOYS10
7443 AD1609 LDA PDLO
7446 C933 CMP #51
7448 B009 ^7453 BCS JS1
744A A560 LDA JSDATA
744C 29EF AND #%11101111
744E 8560 STA JSDATA
7450 4C5D74 JMP YAXIS
7453 C9B2 JS1 CMP #178
    
```



## UTILITIES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 72  
J:5200 .A65

```
7455 9006 ^745D      BCC      YAXIS
7457 A560            LDA      JSDATA
7459 297F            AND      #01111111
745B 8560            STA      JSDATA
745D AD1709          YAXIS     LDA      PDL1
7460 C933            CMP      #51
7462 B009 ^746D      BCS      JS2
7464 A560            LDA      JSDATA
7466 29FB            AND      #11111011
7468 8560            STA      JSDATA
746A 4C7774          JME      JSJAM
746D C9B2            JS2      CMP      #178
746F 9006 ^7477      BCC      JSJAM
7471 A560            LDA      JSDATA
7473 29F7            AND      #11110111
7475 8560            STA      JSDATA
7477 AD0FE8          JSJAM     LDA      SKSTAT
747A 2908            AND      #00001000
747C D006 ^7484      BNE      JSJAM1
747E A560            LDA      JSDATA
7480 29DF            AND      #11011111
7482 8560            STA      JSDATA
7484 AD10C0          JSJAM1    LDA      $C010
7487 D006 ^748F      BNE      JSJAM2
7489 A560            LDA      JSDATA
748B 29DF            AND      #11011111
748D 8560            STA      JSDATA
748F 60              JSJAM2    RTS
7490                JOYS90
7490 A9F7            LDA      #$F7
7492 8560            STA      JSDATA      ; FIRE DOWN WHEN OUT OF FUEL
7494 A59B            LDA      BUCKY      ; SEE IF ON GROUND
7496 C978            CMP      #120
7498 B001 ^749B      BCS      JOY100      ; TIME TO DIE
749A 60              RTS
749B                JOY100
749B 4C6765          JMP      DEADBK

;
; ROUTINE FOR INC/DEC WORD AT 0,1 AND 2,3
;
749E                INC01
749E E600            INC      0
74A0 D002 ^74A4      BNE      INC010
74A2 E601            INC      1
74A4                INC010
74A4 A501            LDA      1      ; GET MSB TO A
74A6 60              RTS
74A7                INC23
74A7 E602            INC      2
74A9 D002 ^74AD      BNE      INC210
74AB E603            INC      3
74AD                INC210
74AD A503            LDA      3      ; MSB TO A
74AF 60              RTS
LIST                L
```

# SOUND TABLES

74B0		SNDRBL			
74B0	0B75	DW	CRASH-1	:	ALIEN DIED
74B2	3575	DW	BEIRE-1	:	BUCK FIRE
74B4	DC74	DW	BEXPI-1	:	BUCK DIED
74B6	5175	DW	HEIRE-1	:	HARPER FIRE
74B8	5E75	DW	PFIRE-1	:	POLE FIRE
74BA	2775	DW	RICOCH-1	:	RICOCHET SOUND (SHOT BOUNCE OFF ZORBA)
74BC	6B75	DW	FUELST-1	:	FUELOUT
74BE	4375	DW	BEIRWN-1	:	WHITE NOISE W/ BUCK SHOT
74C0	7675	DW	ZORBD-1	:	ZORBA DEAD
74C2	B375	DW	PXLPAS-1	:	PASS THRU POLE
74C4	C574	DW	BONNS-1	:	FRE SHIP

74C6	01AE303000	BONUS	DB	1, \$AF, \$30, \$30, \$0, \$30, \$30, \$0, \$30, \$30, 0
------	------------	-------	----	--

74D1	0203030101	FRECHN	DB	2, 3, 3, 1, 1, 1, 2, 3, 2, 2, 2
------	------------	--------	----	---------------------------------

74DD	018E80FF10		DB	1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58
74EB	6068707880		DB	\$60, \$68, \$70, \$78, \$80, \$88, \$90, \$98, \$A0, 1, \$8D, \$A8, \$B0, \$B8, 1, \$8B, \$C0, \$C8
74ED	D00189D801		DB	\$D0, 1, \$89, \$D8, 1, \$87, \$E0, 1, \$85, \$E8, 1, \$83, \$F0, \$FF, 0
750C		CRASH		
750C	018FF0E040		DB	1, \$8F, \$F0, \$E0, \$40, \$50, \$60, \$40, \$70, \$80, \$90, \$A0, \$B0, \$C0
751A	95018A8060		DB	\$95, \$1, \$8A, \$80, \$60, \$55, \$40, \$45, \$30, \$35, \$25, \$20, \$15, \$0

7528		RICOCH		
7528	01AA401004		DB	1, \$AA, \$40, \$10, \$04, \$20, \$28, \$30, \$38, \$40, \$48, \$58, \$70, 0
7536		BFIRE		
7536	01AA808890		DB	1, \$AA, \$80, \$88, \$90, \$98, \$A0, \$A8, \$B0, \$B8, \$C0, \$C8, \$D0, 0
7544		BEIRWN		
7544	0184081018		DB	1, \$84, \$08, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58, 0
7552		HEIRE		
7552	01AA101810		DB	1, \$AA, \$10, \$18, \$10, \$18, \$10, \$18, \$10, \$18, \$10, \$18, 0
755F		PFIRE		
755F	01AA405240		DB	1, \$AA, \$40, \$52, \$40, \$52, \$40, \$52, \$40, \$52, \$40, \$52, 0
756C		FUELST		
756C	01AA807870		DB	1, \$AA, \$80, \$78, \$70, \$68, \$60, \$58, \$50, \$48, 0

7577		ZORBD		
7577	018F80FF10		DB	1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58
7585	018F80FF10		DB	1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58
7593	6068707880		DB	\$60, \$68, \$70, \$78, \$80, \$88, \$90, \$98, \$A0, 1, \$8D, \$A8, \$B0, \$B8, 1, \$8B, \$C0, \$C8
75A5	D00189D801		DB	\$D0, 1, \$89, \$D8, 1, \$87, \$E0, 1, \$85, \$E8, 1, \$83, \$F0, \$FF, 0
75B4	01841E1D1C	PXLPAS	DB	1, \$84, \$1F, \$1D, \$1C, \$1B, \$1A, \$10, \$F, \$E, \$D, \$C, \$B, \$A, \$9, \$8, \$7, \$6, \$5, 0

## PATH TABLES FOR OBJECTS MOVING

SPECIAL CONFIGURATION COMMANDS ARE EQUATED FOR READING TABLE

= 0080	JUMP	EQU	\$80	: JUMP IN TABLE
= 0081	LOADPC	EQU	\$81	: LOAD PC TYPE IMMEDIATE
= 0082	KILL	EQU	\$82	: CANCEL
= 0083	CALL	EQU	\$83	: CALL IN TABLE (1 LEVEL)
= 0084	RET	EQU	\$84	: RETURN FROM CALL
= 0085	SETSPD	EQU	\$85	: SET SPEED

## DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 74  
J:5200 .A65

= 0086 POLPAS EQU \$86 ; POLES EAST BUCK CHECK FOR BUCK BETWEEN  
= 0087 MTRBAK EQU \$87 ; MOTHER SHIP TAKE RETURN PATH  
; TABLE POINTING TO PATHS  
;

75C8 PIHTBL  
75C8 1576 DW PATH0-1 ; BUCK SHOTS  
75CA 2876 DW PATH1-1 ; LEFT POLES  
75CC 6076 DW PATH2-1 ; RIGHT POLES  
75CE 9A76 DW PATH3-1 ; SAUCER PATH#1  
75D0 A276 DW PATH4-1 ; SAUCER PATH#2  
75D2 AA76 DW PATH5-1 ; SAUCER PATH#3  
75D4 B276 DW PATH6-1 ; SAUCER PATH#4  
75D6 B876 DW PATH7-1 ; EXPLOSION PATH  
75D8 D276 DW PATH8-1 ; CANCEL OBJECT  
75DA C076 DW PATH9-1 ; BUCK EXPLOSION PATH  
75DC D376 DW PATH10-1 ; HOPPER PATHS 1-2  
75DE F576 DW PATH11-1  
75E0 1777 DW PATH12-1 ; ENEMY SHOT PATH  
75E2 2477 DW PATH13-1 ; POLE RAY LEFT  
75E4 2977 DW PATH14-1 ; POLE RAY RIGHT  
75E6 2E77 DW PATH15-1 ; MOTHER ZORBA ENTRANCE PATH#1  
75E8 6777 DW PATH16-1 ; MOTHER ZORBA ENTRANCE PATH#2  
75EA A077 DW PATH17-1 ; MOTHER ZORBA RETURN PATH 1  
75EC B777 DW PATH18-1 ; MOTHER ZORBA RETURN PATH 2  
75EE CE77 DW PATH19-1 ; STAR PATHS 19-30  
75F0 D377 DW PATH20-1  
75F2 DA77 DW PATH21-1  
75F4 E177 DW PATH22-1  
75F6 E677 DW PATH23-1  
75F8 ED77 DW PATH24-1  
75FA F477 DW PATH25-1  
75FC F977 DW PATH26-1  
75FE 0078 DW PATH27-1  
7600 0778 DW PATH28-1  
7602 0C78 DW PATH29-1  
7604 1378 DW PATH30-1  
7606 1A78 DW PATH31-1 ; RICOCHET SHOTS #'S 31-38  
7608 2978 DW PATH32-1  
760A 3878 DW PATH33-1  
760C 4778 DW PATH34-1  
760E 5678 DW PATH35-1  
7610 5B78 DW PATH36-1  
7612 6A78 DW PATH37-1  
7614 7978 DW PATH38-1

; PATH0 IS PATH FOR BUCK SHOTS

7616 PATH0  
7616 184083 DB 24, \$40, CALL  
7618 2276 DW SUB0  
761A 83 DB CALL  
761C 2276 DW SUB0  
761E 83 DB CALL  
761F 2276 DW SUB0  
7621 82 DB KILL  
7622 SUB0  
7622 1072044004 DB 16, \$72, 4, \$40, 4, \$E8, RET

; PATH1, PATH2 FOR POLES

DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 75  
J:5200 .A65

PATH1

7629	0264	DB	2,\$64
762B	016401C403	DB	1,\$64,1,\$C4,3,\$44,1,\$C4
7633	83	DB	CALL
7634	B179	DW	SUB1
7636	83	DB	CALL
7637	B179	DW	SUB1
7639	83	DB	CALL
763A	B179	DW	SUB1
763C	0244	DB	2,\$44
763E	016401C483	DB	1,\$64,1,\$C4,CALL
7643	B879	DW	SUB3
7645	0244	DB	2,\$44
7647	014401C483	DB	1,\$44,1,\$C4,CALL
764C	B179	DW	SUB1
764E	83	DB	CALL
764F	BC79	DW	SUB4
7651	83	DB	CALL
7652	B179	DW	SUB1
7654	83	DB	CALL
7655	B879	DW	SUB3
7657	86	DB	POLEAS
7658	83	DB	CALL
7659	B179	DW	SUB1
765B	83	DB	CALL
765C	B879	DW	SUB3
765E	28C482	DB	40,\$C4,KILL

: CHECK BACK BETWEEN

PATH2

7661	0264	DB	2,\$64
7663	016401CC03	DB	1,\$64,1,\$CC,3,\$44,1,\$CC
766B	83	DB	CALL
766C	C179	DW	SUB5
766E	83	DB	CALL
766F	C179	DW	SUB5
7671	83	DB	CALL
7672	C179	DW	SUB5
7674	0244	DB	2,\$44
7676	016401CC83	DB	1,\$64,1,\$CC,CALL
767B	C879	DW	SUB7
767D	0244	DB	2,\$44
767F	014401CC83	DB	1,\$44,1,\$CC,CALL
7684	C179	DW	SUB5
7686	83	DB	CALL

## DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 76  
J:5200 .A65

7687	CC79	DW	SUB8	
7689	83	DB	CALL	
768A	C179	DW	SUB5	
768C	83	DB	CALL	
768D	C879	DW	SUB7	
768F	86	DB	POLPAS	: CHECK BUCK BETWEEN POLES
7690	83	DB	CALL	
7691	C179	DW	SUB5	
7693	83	DB	CALL	
7694	C879	DW	SUBZ	
769A	28C82	DB	40,\$CC,KILL	
: PATH 3-6 ARE SAUCERS				
: PATH3				
7699	7040	DB	112,\$40	
769B	83	DB	CALL	
769C	C179	DW	SUB9	: ENTRANCE
769E	83	DB	CALL	
769F	7879	DW	SUB10	: EXIT
: PATH4				
76A1	7040	DB	112,\$40	
76A3	83	DB	CALL	
76A4	C179	DW	SUB9	: ENTRANCE
76A6	83	DB	CALL	
76A7	A079	DW	SUB12	: EXIT
: PATH5				
76A9	7040	DB	112,\$40	
76AB	83	DB	CALL	
76AC	8979	DW	SUB11	: ENTRANCE
76AE	83	DB	CALL	
76AF	A079	DW	SUB12	: EXIT
: PATH6				
76B1	7040	DB	112,\$40	
76B3	83	DB	CALL	
76B4	8979	DW	SUB11	: ENTRANCE
76B6	83	DB	CALL	
76B7	7879	DW	SUB10	: EXIT
: PATH7				
76B9	8107	DB	LOADPC,7	: SET AS EXPLOSION
76BB	8510	DB	SETSPD,\$10	: SET SPEED
76BD	0500	DB	5,0	: CYCLE THROUGH SIZES AND STOP
76BF	0520	DB	5,\$20	
: PATH9				
76C1	8510	DB	SETSPD,\$10	
76C3	05A8	DB	5,\$A8	
76C5	0522	DB	5,\$22	
76C7	0522	DB	5,\$22	
76C9	05AA	DB	5,\$AA	
76CB	0522	DB	5,\$22	
76CD	0522	DB	5,\$22	
76CF	0522	DB	5,\$22	



76C2	0522	DB	5,\$AA
76CB	0522	DB	5,\$22
76CD	0522	DB	5,\$22
76CF	0522	DB	5,\$22

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 77  
J:5200 .A65

DATA TABLES

76D1	0522	DB	5,\$22
------	------	----	--------

76D3		PATH8	
76D3	82	DB	KILL : CANCEL

: HOPPER PATHS

76D4		PATH10	: PATH10 FOR BACK/FORTH LEFT TO RIGHT
------	--	--------	---------------------------------------

76D4	04C404E401	DB	4,\$C4,4,\$E4,1,\$20,CALL
76DB	5279	DW	SUB22
76DD	10C4012083	DB	16,\$C4,1,\$20,CALL
76E2	C778	DW	SUB16
76E4	10CC08EC01	DB	16,\$CC,8,\$EC,1,\$20,CALL
76EB	1479	DW	SUB20
76ED	20C4012083	DB	32,\$C4,1,\$20,CALL
76F2	8978	DW	SUB14
76E4	7ECC	DB	127,\$CC

76E6		PATH11	: BACKWARDS PATH10
------	--	--------	--------------------

76E6	04CC04EC01	DB	4,\$CC,4,\$EC,1,\$20,CALL
76ED	E678	DW	SUB17
76FF	10CC012083	DB	16,\$CC,1,\$20,CALL
7704	3379	DW	SUB21
7706	10E408E401	DB	16,\$C4,8,\$E4,1,\$20,CALL
770D	A878	DW	SUB15
770E	20CC012083	DB	32,\$CC,1,\$20,CALL
7714	E578	DW	SUB19
7716	7FC4	DB	127,\$C4

7718		PATH12	: ENEMY SHOTS
------	--	--------	---------------

7718	83	DB	CALL
7719	1E77	DW	SUB23
771B	80	DB	JUMP
771C	1877	DW	PATH12

771E		SUB23	
771E	04C4046408	DB	4,\$C4,4,\$64,8,\$64,RET

7725		PATH13	: LEFT POLE RAY
------	--	--------	-----------------

7725	7F8080	DB	127,\$80,JUMP
7728	2577	DW	PATH13

772A		PATH14	: RIGHT POLE RAY
------	--	--------	------------------

772A	7F8880	DB	127,\$88,JUMP
772B	2A77	DW	PATH14

772E		PATH15	: ZORBA ENTRANCE PATH #1
------	--	--------	--------------------------

772E	0840	DB	8,\$40
7731	10C8	DB	16,\$C8

7733	10BA	DB	16,\$BA
7735	0840	DB	8,\$40
7737	08C0	DB	8,\$C0
7739	4880	DB	72,\$80
773B	0840	DB	8,\$40

773D	10B2	DB	16,\$B2
773F	0840	DB	8,\$40
7741	08C8	DB	8,\$C8

7743	48BA	DB	72,\$BA
7745	08C8	DB	8,\$C8

7747	0450	DB	4,\$50
7749	04C0	DB	4,\$C0

774B	10A2	DB	16,\$A2
------	------	----	---------

DATA TABLES

1	774D	02C0	DB	2,\$C0	
2	774F	02FA	DB	2,\$FA	
3	7751	0880	DB	8,\$80	
4	7753	02C8	DB	2,\$C8	
5					
6	7755	0272	DB	2,\$72	
7	7757	0280	DB	2,\$80	
8	7759	02C0	DB	2,\$C0	
9					
10	775B	02EA	DB	2,\$EA	
11	775D	0280	DB	2,\$80	
12					
13	775F	02E2	DB	2,\$E2	
14	7761	0888	DB	8,\$88	
15	7763	0880	DB	8,\$80	
16	7765	0888	DB	8,\$88	
17					
18	7767	87	DB	MTRBAK	: MOTHER ZORBA COME BACK NOW
19					
20	7768				: ZORBA ENTRANCE PATH #2
21	7768	0840	DB	8,\$40	
22	776A	10C0	DB	16,\$C0	
23					
24	776C	10B2	DB	16,\$B2	
25	776E	0840	DB	8,\$40	
26	7770	08C8	DB	8,\$C8	
27	7772	4888	DB	72,\$88	
28	7774	0840	DB	8,\$40	
29					
30	7776	10BA	DB	16,\$BA	
31	7778	0840	DB	8,\$40	
32	777A	08C0	DB	8,\$C0	
33					
34	777C	48B2	DB	72,\$B2	
35	777E	08C0	DB	8,\$C0	
36					
37	7780	0450	DB	4,\$50	
38	7782	04C8	DB	4,\$C8	
39					
40	7784	10AA	DB	16,\$AA	
41	7786	02C8	DB	2,\$C8	
42					
43	7788	02F2	DB	2,\$F2	
44	778A	0888	DB	8,\$88	
45	778C	02C0	DB	2,\$C0	
46					
47	778E	0272	DB	2,\$72	
48	7790	0288	DB	2,\$88	
49	7792	02C8	DB	2,\$C8	
50					
51	7794	02E2	DB	2,\$E2	
52	7796	0288	DB	2,\$88	
53					
54	7798	02FA	DB	2,\$FA	
55	779A	0880	DB	8,\$80	
56	779C	0888	DB	8,\$88	
57	779E	0880	DB	8,\$80	
	77A0	87	DB	MTRBAK	: MOTHER ZORBA COME BACK NOW

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 79

J:5200 .A65

## DATA TABLES

77A1 PATH17 ; MOTHER ZORBA RETURN PATH #1

77A1 02EB DB 2,\$EC

77A3 10A0 DB 16,\$A0

77A5 04ED DB 4,\$ED

77A7 1088 DB 16,\$88

77A9 04E4 DB 4,\$E4

77AB 2080 DB 32,\$80

77AD 08ED DB 8,\$ED

77AF 20A8 DB 32,\$A8

77B1 08E4 DB 8,\$E4

77B3 40A0 DB 64,\$A0

77B5 7FDD DB 127,\$DD

77B7 82 DB KILL

77B8 PATH18 ; MOTHER ZORBA RETURN PATH #2

77B8 02E4 DB 2,\$E4

77BA 10A8 DB 16,\$A8

77BC 04F5 DB 4,\$E5

77BE 1080 DB 16,\$80

77C0 04EC DB 4,\$EC

77C2 2088 DB 32,\$88

77C4 08E5 DB 8,\$E5

77C6 20A0 DB 32,\$A0

77C8 08EC DB 8,\$EC

77CA 40A8 DB 64,\$A8

77CC 7ED5 DB 127,\$D5

77CE 82 DB KILL

; STAR PATHS 19-30

77CF PATH12 ; STRAIGHT UP

77CE 7E4080 DB 127,\$40,JUMP

77D2 CF77 DW PATH19

77D4 PATH20 ; 1:2

77D4 014001C880 DB 1,\$40,1,\$C8,JUMP

77D2 D477 DW PATH20

77D8 PATH21 ; 2:1

77D8 018801C880 DB 1,\$88,1,\$C8,JUMP

77E0 DB77 DW PATH21

77E2 PATH22 ; RIGHT

77E2 7E8880 DB 127,\$88,JUMP

77E5 E277 DW PATH22

77E7 PATH23

DATA TABLES

77E7	018801CC80	DB	1,\$88,1,\$CC,JUMP
77EC	E777	DW	PATH23
77EE	014401CC80	DB	1,\$44,1,\$CC,JUMP
77E3	EE77	DW	PATH24
77E5			PATH25
77E5	7E4480	DB	127,\$44,JUMP : DOWN
77F8	F577	DW	PATH25
77FA			PATH26
77FA	014401C480	DB	1,\$44,1,\$C4,JUMP
77EF	EA77	DW	PATH26
7801			PATH27
7801	018001C480	DB	1,\$80,1,\$C4,JUMP
7806	0178	DW	PATH27
7808			PATH28
7808	7E8080	DB	127,\$80,JUMP : LEFT
780B	0878	DW	PATH28
780D			PATH29
780D	018001C080	DB	1,\$80,1,\$C0,JUMP
7812	0D78	DW	PATH29
7814			PATH30
7814	014001C080	DB	1,\$40,1,\$C0,JUMP
7819	1478	DW	PATH30
			: RICOCHET SHOT PATHS
781B			PATH31
781B	810483	DB	LOADPC,4,CALL
781E	2378	DW	SUB24
7820	80	DB	JUMP
7821	1B78	DW	PATH31
7823			SUB24
7823	04C0046008	DB	4,\$C0,4,\$60,8,\$60,RET
782A			PATH32
782A	810483	DB	LOADPC,4,CALL
782D	3278	DW	SUB25
782E	80	DB	JUMP
7830	2A78	DW	PATH32
7832			SUB25
7832	04C004E808	DB	4,\$C0,4,\$E8,8,\$E8,RET
7839			PATH33
7839	810483	DB	LOADPC,4,CALL
783C	4178	DW	SUB26
783E	80	DB	JUMP
783F	8978	DW	PATH33
7841			SUB26
7841	048804A808	DB	4,\$88,4,\$A8,8,\$A8,RET
7848			PATH34
7848	810483	DB	LOADPC,4,CALL
784B	5078	DW	SUB27
784D	80	DB	JUMP
784E	4878	DW	PATH34
7850			SUB27
7850	04CC04EC08	DB	4,\$CC,4,\$EC,8,\$EC,RET
7857			PATH35
7857	810480	DB	LOADPC,4,JUMP
785A	1877	DW	PATH12
785C			PATH36
785C	810483	DB	LOADPC,4,CALL
785F	6478	DW	SUB28

785C	810483	DB	LOADPC, 4, CALL
785F	6478	DW	SUB28

PATH36

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 81  
J:5200 .A65

DATA TABLES

7861	80	DB	JUMP
7862	5C78	DW	PATH36
7864		SUB28	
7864	04C404E408	DB	4, \$C4, 4, \$E4, 8, \$E4, RET
786B		PATH37	
786B	810483	DB	LOADPC, 4, CALL
786E	7378	DW	SUB29
7870	80	DB	JUMP
7871	6B78	DW	PATH37
7873		SUB29	
7873	048004A008	DB	4, \$80, 4, \$A0, 8, \$A0, RET
787A		PATH38	
787A	810483	DB	LOADPC, 4, CALL
787D	8278	DW	SUB30
787F	80	DB	JUMP
7880	7A78	DW	PATH38
7882		SUB30	
7882	04C004E408	DB	4, \$C0, 4, \$E4, 8, \$E4, RET
7889		SUB14	
7889	04C80A4009	DB	4, \$C8, 10, \$40, 9, \$C8, 4, \$40, 11, \$C8, 3, \$88, 2, \$C8, 7, \$A8
7899	02CC03880B	DB	2, \$CC, 3, \$88, 11, \$CC, 4, \$44, 9, \$CC, 10, \$44, 4, \$CC, RET
78A8		SUB15	
78A8	03C8044005	DB	3, \$C8, 4, \$40, 5, \$C8, 4, \$40, 2, \$C8, 2, \$40, 9, \$C8, 9, \$A8
78B8	09CC024402	DB	9, \$CC, 2, \$44, 2, \$CC, 4, \$44, 5, \$CC, 4, \$44, 3, \$CC, RET
78C7		SUB16	
78C7	02C8034002	DB	2, \$C8, 3, \$40, 2, \$C8, 2, \$40, 3, \$C8, 2, \$40, 5, \$C8, 6, \$A8
78D7	05CC024403	DB	5, \$CC, 2, \$44, 3, \$CC, 2, \$44, 2, \$CC, 3, \$44, 2, \$CC, RET
78E4		SUB17	
78E4	02C8024004	DB	2, \$C8, 2, \$40, 4, \$C8, 4, \$88, 4, \$CC, 2, \$44, 2, \$CC, RET
78E5		SUB19	
78F5	04C00A4009	DB	4, \$C0, 10, \$40, 9, \$C0, 4, \$40, 11, \$C0, 3, \$80, 2, \$C0, 7, \$A0
7905	02C403800B	DB	2, \$C4, 3, \$80, 11, \$C4, 4, \$44, 9, \$C4, 10, \$44, 4, \$C4, RET
7914		SUB20	
7914	03C0044005	DB	3, \$C0, 4, \$40, 5, \$C0, 4, \$40, 2, \$C0, 2, \$40, 9, \$C0, 9, \$A0
7924	09C4024402	DB	9, \$C4, 2, \$44, 2, \$C4, 4, \$44, 5, \$C4, 4, \$44, 3, \$C4, RET
7933		SUB21	
7933	02C4034002	DB	2, \$C4, 3, \$40, 2, \$C0, 2, \$40, 3, \$C0, 2, \$40, 5, \$C0, 6, \$A0
7943	05C4024403	DB	5, \$C4, 2, \$44, 3, \$C4, 2, \$44, 2, \$C4, 3, \$44, 2, \$C4, RET
7952		SUB22	
7952	02C0024004	DB	2, \$C0, 2, \$40, 4, \$C0, 4, \$80, 4, \$C4, 2, \$44, 2, \$C4, RET
7961		SUB9	
7961	10C8	DB	16, \$C8
7963	10C8	DB	16, \$C8
7965	10EA	DB	16, \$EA
7967	1062	DB	16, \$62
7969	1062	DB	16, \$62
796B	1072	DB	16, \$72
796D	1072	DB	16, \$72
796F	1072	DB	16, \$72



DATA TABLES

7971	0872	DB	8,\$72
7972	0862	DB	8,\$62
7975	108084	DB	16,\$90,RET
7978	SUB10		
7978	0864	DB	8,\$64
797A	0875	DB	8,\$75
797C	1075	DB	16,\$75
797E	10FD	DB	16,\$FD
7980	10FD	DB	16,\$FD
7982	10EC	DB	16,\$EC
7984	1064	DB	16,\$64
7986	30E482	DB	48,\$E4,KILL
7989	SUB11		
7989	10E0	DB	16,\$C0
798B	10C0	DB	16,\$C0
798D	10E2	DB	16,\$E2
798F	1062	DB	16,\$62
7991	1062	DB	16,\$62
7993	1072	DB	16,\$72
7995	1072	DB	16,\$72
7997	1072	DB	16,\$72
7999	0872	DB	8,\$72
799B	0862	DB	8,\$62
799D	108884	DB	16,\$88,RET
79A0	SUB12		
79A0	0864	DB	8,\$64
79A2	0875	DB	8,\$75
79A4	1075	DB	16,\$75
79A6	10F5	DB	16,\$F5
79A8	10F5	DB	16,\$F5
79AA	10E4	DB	16,\$E4
79AC	1064	DB	16,\$64
79AE	30EC82	DB	48,\$EC,KILL

79B1 SUB1  
79B1 036401C480

; 2\*4:1 LINE STARTING W/GROW

DB 3,\$64,1,\$C4,JUMP

79B1

SUB1

: 2\*4:1 LINE STARTING W/GROW

79B1 036401C480

DB

3,\$64,1,\$C4,JUMP

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 83

## DATA TABLES

J:5200 .A65

79B6 BC79

DW

SUB4

79B8

SUB2

: 2\*4:1 LINE

79B8 034401C4

DB

3,\$44,1,\$C4

79B8

SUB4

79BC 034401C484

DB

3,\$44,1,\$C4,RET

: 4:1 LINE

79C1

SUB5

: 2\*4:1 LINE STARTING W/GROW

79C1 036401CC80

DB

3,\$64,1,\$CC,JUMP

79C6 CC79

DW

SUB8

79C8

SUB7

: 2\*4:1 LINE

79C8 034401CC

DB

3,\$44,1,\$CC

79CC

SUB8

79CC 034401CC84

DB

3,\$44,1,\$CC,RET

: 4:1 LINE

DATA TABLES

PICTURE TYPE TO PICTURE CODE CONVERSION TABLES  
A NUMBER FROM 0-8 WILL INDEX A GROUP OF 4, AND ANOTHER  
NUMBER FROM 0-3 WILL INDEX ONE OF THE 4 WITHIN THE  
SUB-GROUP. 0 IS LARGEST WHILE 8 IS SMALLEST.

79D1	E3F70B1E33	DCODEL	DB	LOW TPOLE,LOW TSAUC,LOW TDROID,LOW TESHT,LOW TSHOT,LOW TZORBA,LOW TSHOT,LOW TPSHT,LOW TSTAR
79DA	79797A7A7A	DCODEH	DB	HIGH TPOLE,HIGH TSAUC,HIGH TDROID,HIGH TESHT,HIGH TSHOT,HIGH TZORBA,HIGH TSHOT,HIGH TPSHT,HI
GH TSTAR				
U 79E3	0101F7F7ED	TPOLE	DB	LOW POL6,LOW POL6,LOW POL5,LOW POL5,LOW POL4,LOW POL3,LOW POL2,LOW POL1,LOW DOT,LOW DOT
U 79ED	979Z969696		DB	HIGH POL6,HIGH POL6,HIGH POL5,HIGH POL5,HIGH POL4,HIGH POL3,HIGH POL2,HIGH POL1,HIGH DOT,HIG
H DOT				
U 79F7	0000000000	TSAUC	DB	LOW SAU4,LOW SAU4,LOW SAU3,LOW SAU3,LOW SAU2,LOW SAU2,LOW SAU1,LOW SAU1,LOW DOT,LOW DOT
U 7A01	0000000000		DB	HIGH SAU4,HIGH SAU4,HIGH SAU3,HIGH SAU3,HIGH SAU2,HIGH SAU2,HIGH SAU1,HIGH SAU1,HIGH DOT,HIG
H DOT				
U 7A0B	0000000000	TDROID	DB	LOW HOP5,LOW HOP4,LOW HOP3,LOW HOP3,LOW HOP2,LOW HOP2,LOW HOP1,LOW HOP1,LOW DOT,LOW DOT
U 7A15	0000000000		DB	HIGH HOP5,HIGH HOP4,HIGH HOP3,HIGH HOP3,HIGH HOP2,HIGH HOP2,HIGH HOP1,HIGH HOP1,HIGH DOT,HIG
H DOT				
U 7A1F	0000000000	TESHT	DB	LOW SHOT3,LOW SHOT2,LOW SHOT1,LOW SHOT1,LOW SHOT1,LOW SHOT1,LOW SHOT1,LOW SHOT1,LOW SHOT1,LO
W DOT				
U 7A29	0000000000		DB	HIGH SHOT3,HIGH SHOT2,HIGH SHOT1,HIGH SHOT1,HIGH SHOT1,HIGH SHOT1,HIGH SHOT1,HIGH SHOT1,HIGH
SHOT1,HIGH DOT				
U 7A33	0000000000	TSHOT	DB	LOW SHOT6,LOW SHOT6,LOW SHOT6,LOW SHOT6,LOW SHOT5,LOW SHOT4,LOW SHOT3,LOW SHOT2,LOW SHOT1,LO
W DOT				
U 7A3D	0000000000		DB	HIGH SHOT6,HIGH SHOT6,HIGH SHOT6,HIGH SHOT6,HIGH SHOT5,HIGH SHOT4,HIGH SHOT3,HIGH SHOT2,HIGH
SHOT1,HIGH DOT				
U 7A47	0000000000	TZORBA	DB	LOW ZOR5,LOW ZOR5,LOW ZOR4,LOW ZOR4,LOW ZOR3,LOW ZOR3,LOW ZOR2,LOW ZOR2,LOW ZOR1,LOW DOT
U 7A51	0000000000		DB	HIGH ZOR5,HIGH ZOR5,HIGH ZOR4,HIGH ZOR4,HIGH ZOR3,HIGH ZOR3,HIGH ZOR2,HIGH ZOR2,HIGH ZOR1,HI
GH DOT				
U 7A5B	0000000000	TPSHT	DB	LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LOW SHOT3,LO
W SHOT3				
U 7A65	0000000000		DB	HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH SHOT3,HIGH
SHOT3,HIGH SHOT3				
U 7A6F	0000000000	TSTAR	DB	LOW STER,LOW STER,LOW STER,LOW STER,LOW STER,LOW STER,LOW STER,LOW STER,LOW STER,LOW STER
U 7A79	0000000000		DB	HIGH STER,HIGH STER,HIGH STER,HIGH STER,HIGH STER,HIGH STER,HIGH STER,HIGH STER,HIGH STER,HI
GH STER				
7A83	95A9BDD1E5	MCODEL	DB	LOW MPOLE,LOW MSAUC,LOW MDROID,LOW MESHT,LOW MSHOT,LOW MZORBA,LOW MSHOT,LOW MPSHT,LOW MSTAR
7A8C	7A7A7A7A7A	MCODEH	DB	HIGH MPOLE,HIGH MSAUC,HIGH MDROID,HIGH MESHT,HIGH MSHOT,HIGH MZORBA,HIGH MSHOT,HIGH MPSHT,HI
GH MSTAR				
U 7A95	0000000000	MPOLE	DB	LOW MPOL6,LOW MPOL6,LOW MPOL5,LOW MPOL5,LOW MPOL4,LOW MPOL3,LOW MPOL2,LOW MPOL1,LOW MDOT,LOW
MDOT				
U 7A9F	0000000000		DB	HIGH MPOL6,HIGH MPOL6,HIGH MPOL5,HIGH MPOL5,HIGH MPOL4,HIGH MPOL3,HIGH MPOL2,HIGH MPOL1,HIGH
MDOT,HIGH MDOT				
U 7AA9	0000000000	MSAUC	DB	LOW MSAU4,LOW MSAU4,LOW MSAU3,LOW MSAU3,LOW MSAU2,LOW MSAU2,LOW MSAU1,LOW MSAU1,LOW MDOT,LOW
MDOT				
U 7AB3	0000000000		DB	HIGH MSAU4,HIGH MSAU4,HIGH MSAU3,HIGH MSAU3,HIGH MSAU2,HIGH MSAU2,HIGH MSAU1,HIGH MSAU1,HIGH
MDOT,HIGH MDOT				
U 7ABD	0000000000	MDROID	DB	LOW MHOP5,LOW MHOP4,LOW MHOP3,LOW MHOP3,LOW MHOP2,LOW MHOP2,LOW MHOP1,LOW MHOP1,LOW MDOT,LOW
MDOT				
U 7AC7	0000000000		DB	HIGH MHOP5,HIGH MHOP4,HIGH MHOP3,HIGH MHOP3,HIGH MHOP2,HIGH MHOP2,HIGH MHOP1,HIGH MHOP1,HIGH

U 7ABD	0000000000	MDROID	DB	LOW MHOP5,LOW MHOP4,LOW MHOP3,LOW MHOP3,LOW MHOP2,LOW MHOP2,LOW MHOP1,LOW MHOP1,LOW MDOT,LOW MDOT
U 7AC7	0000000000		DB	HIGH MHOP5,HIGH MHOP4,HIGH MHOP3,HIGH MHOP3,HIGH MHOP2,HIGH MHOP2,HIGH MHOP1,HIGH MHOP1,HIGH MDOT,HIGH MDOT
U 7AD1	0000000000	MESHT	DB	LOW MSHOT3,LOW MSHOT2,LOW MDOT,LOW MDOT,LOW MDOT,LOW MDOT,LOW MDOT,LOW MDOT,LOW MDOT,LOW MDOT
U 7ADB	0000000000		DB	HIGH MSHOT3,HIGH MSHOT2,HIGH MDOT,HIGH MDOT,HIGH MDOT,HIGH MDOT,HIGH MDOT,HIGH MDOT,HIGH MDOT,HIGH MDOT
U 7AE5	0000000000	MSHOT	DB	LOW MSHOT6,LOW MSHOT6,LOW MSHOT6,LOW MSHOT6,LOW MSHOT5,LOW MSHOT4,LOW MSHOT3,LOW MSHOT2,LOW MSHOT1,LOW MDOT
U 7AEF	0000000000		DB	HIGH MSHOT6,HIGH MSHOT6,HIGH MSHOT6,HIGH MSHOT6,HIGH MSHOT5,HIGH MSHOT4,HIGH MSHOT3,HIGH MSHOT2,HIGH MSHOT1,HIGH MDOT
U 7AF9	0000000000	MZORBA	DB	LOW MZOR5,LOW MZOR5,LOW MZOR4,LOW MZOR4,LOW MZOR3,LOW MZOR3,LOW MZOR2,LOW MZOR2,LOW MZOR1,LOW MDOT
U 7B03	0000000000		DB	HIGH MZOR5,HIGH MZOR5,HIGH MZOR4,HIGH MZOR4,HIGH MZOR3,HIGH MZOR3,HIGH MZOR2,HIGH MZOR2,HIGH MZOR1,HIGH MDOT
U 7B0D	0000000000	MPSHT	DB	LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3,LOW MSHOT3
U 7B17	0000000000		DB	HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3,HIGH MSHOT3
U 7B21	0000000000	MSTAR	DB	LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER,LOW MSTER
U 7B2B	0000000000		DB	HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER,HIGH MSTER

PATTERN TABLES FOR SCROLLING

EACH DATA BYTE REPRESENTS A ROW THAT CONTAINS A STRIPE.  
255 ENTRIES FORCE BOARDER COLOR USED FOR SHORT ROWS.

7B35	0606070707	SCR0L0	DB	\$06,\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07
7B44	0707070706		DB	\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06
7B54	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07
7B64	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7B74	0706060607		DB	\$07,\$06,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
	= 0050	SCRLEN	EQU	*-SCR0L0
7B85	0606070707	SCR0L1	DB	\$06,\$06,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07
7B94	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7BA4	0606070707		DB	\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7BB4	0606070707		DB	\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7BC4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7BD5	0607060707	SCR0L2	DB	\$06,\$07,\$06,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07
7BE4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7BF4	0707060607		DB	\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7C04	0707070606		DB	\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7C14	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06,\$06,\$07,\$07
7C25	0607060707	SCR0L3	DB	\$06,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06
7C34	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07
7C44	0707070706		DB	\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7C54	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7C64	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06,\$06
7C75	0607070607	SCR0L4	DB	\$06,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07
7C84	0706070707		DB	\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07
7C94	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07
7CA4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06,\$07,\$07,\$07
7CB4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
7CC5	0607070706	SCR0L5	DB	\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07
7CD4	0707070607		DB	\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07
7CE4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07
7CF4	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06
7D04	0707070707		DB	\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07
	= 7D15	SCREND	EQU	*
7D15	0000000001	04	DB	\$00,\$00,\$00,\$00,\$01,\$01,\$01,\$01,\$02,\$02,\$02,\$02,\$03,\$03,\$03,\$03
7D25	0404040405		DB	\$04,\$04,\$04,\$04,\$05,\$05,\$05,\$05,\$06,\$06,\$06,\$06,\$07,\$07,\$07,\$07
7D35	0808080809		DB	\$08,\$08,\$08,\$08,\$09,\$09,\$09,\$09,\$0A,\$0A,\$0A,\$0A,\$0B,\$0B,\$0B,\$0B
7D45	0C0C0C0C0D		DB	\$0C,\$0C,\$0C,\$0C,\$0D,\$0D,\$0D,\$0D,\$0E,\$0E,\$0E,\$0E,\$0F,\$0F,\$0F,\$0F
7D55	1010101011		DB	\$10,\$10,\$10,\$10,\$11,\$11,\$11,\$11,\$12,\$12,\$12,\$12,\$13,\$13,\$13,\$13
7D65	1414141415		DB	\$14,\$14,\$14,\$14,\$15,\$15,\$15,\$15,\$16,\$16,\$16,\$16,\$17,\$17,\$17,\$17
7D75	1818181819		DB	\$18,\$18,\$18,\$18,\$19,\$19,\$19,\$19,\$1A,\$1A,\$1A,\$1A,\$1B,\$1B,\$1B,\$1B
7D85	1C1C1C1C1D		DB	\$1C,\$1C,\$1C,\$1C,\$1D,\$1D,\$1D,\$1D,\$1E,\$1E,\$1E,\$1E,\$1F,\$1F,\$1F,\$1F
7D95	2020202021		DB	\$20,\$20,\$20,\$20,\$21,\$21,\$21,\$21,\$22,\$22,\$22,\$22,\$23,\$23,\$23,\$23
7DA5	2424242425		DB	\$24,\$24,\$24,\$24,\$25,\$25,\$25,\$25,\$26,\$26,\$26,\$26,\$27,\$27,\$27,\$27
7DB5	E8E8E8E8E9		DB	\$E8,\$E8,\$E8,\$E8,\$E9,\$E9,\$E9,\$E9,\$EA,\$EA,\$EA,\$EA,\$EB,\$EB,\$EB,\$EB
7DC5	ECCECECECE		DB	\$EC,\$EC,\$EC,\$EC,\$ED,\$ED,\$ED,\$ED,\$EE,\$EE,\$EE,\$EE,\$EF,\$EF,\$EF,\$EF
7DD5	EOEOEOEOE1		DB	\$EO,\$EO,\$EO,\$EO,\$E1,\$E1,\$E1,\$E1,\$E2,\$E2,\$E2,\$E2,\$E3,\$E3,\$E3,\$E3
7DE5	F4F4F4F4F5		DB	\$F4,\$F4,\$F4,\$F4,\$F5,\$F5,\$F5,\$F5,\$F6,\$F6,\$F6,\$F6,\$F7,\$F7,\$F7,\$F7
7DF5	F8F8F8F8F9		DB	\$F8,\$F8,\$F8,\$F8,\$F9,\$F9,\$F9,\$F9,\$FA,\$FA,\$FA,\$FA,\$FB,\$FB,\$FB,\$FB



7DB5	E8E8E8E8E9	DB	\$E8,\$E8,\$E8,\$E8,\$E9,\$E9,\$E9,\$E9,\$EA,\$EA,\$EA,\$EA,\$EB,\$EB,\$EB,\$EB
7DC5	ECCECECECED	DB	\$EC,\$EC,\$EC,\$EC,\$ED,\$ED,\$ED,\$ED,\$EE,\$EE,\$EE,\$EE,\$EF,\$EF,\$EF,\$EF
7DD5	F0E0E0E0E1	DB	\$E0,\$E0,\$E0,\$E0,\$E1,\$E1,\$E1,\$E1,\$E2,\$E2,\$E2,\$E2,\$E3,\$E3,\$E3,\$E3
7DE5	F4F4F4F4F5	DB	\$F4,\$F4,\$F4,\$F4,\$F5,\$F5,\$F5,\$F5,\$F6,\$F6,\$F6,\$F6,\$F7,\$F7,\$F7,\$F7
7DF5	F8F8F8F8F9	DB	\$F8,\$F8,\$F8,\$F8,\$F9,\$F9,\$F9,\$F9,\$FA,\$FA,\$FA,\$FA,\$FB,\$FB,\$FB,\$FB

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 86  
J:5200 .A65

# DATA TABLES

7E05	ECCECECECED	DB	\$EC,\$EC,\$EC,\$EC,\$ED,\$ED,\$ED,\$ED,\$EE,\$EE,\$EE,\$EE,\$EF,\$EF,\$EF,\$EF
7E15	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E25	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E35	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E45	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E55	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E65	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E75	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E85	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7E95	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7EA5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7EB5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7EC5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7ED5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7EE5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7EF5	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06
7F05	0002040600	DB	\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06,\$00,\$02,\$04,\$06

MUST BE ON 1K BOUNDARY  
QWAZZO EQU HIGH QWAZZO\*100  
ORG

7F15	P1L		
7F15	F018406890	DB	\$F0,\$18,\$40,\$68,\$90,\$B8,\$E0,\$08,\$30,\$58,\$80,\$A8,\$D0,\$F8,\$20,\$48
7F25	7098C0E810	DB	\$70,\$98,\$C0,\$E8,\$10,\$38,\$60,\$88,\$B0,\$D8,\$00,\$28,\$50,\$78,\$A0,\$C8
7F35	F018406890	DB	\$F0,\$18,\$40,\$68,\$90,\$B8,\$E0,\$08,\$30,\$58,\$80,\$A8,\$D0,\$F8,\$20,\$48
7F45	7098C0E810	DB	\$70,\$98,\$C0,\$E8,\$10,\$38,\$60,\$88,\$B0,\$D8,\$00,\$28,\$50,\$78,\$A0,\$C8
7F55	F018406890	DB	\$F0,\$18,\$40,\$68,\$90,\$B8,\$E0,\$08,\$30,\$58,\$80,\$A8,\$D0,\$F8,\$20,\$48
7F65	7098C0E810	DB	\$70,\$98,\$C0,\$E8,\$10,\$38,\$60,\$88,\$B0,\$D8,\$00,\$28,\$50,\$78,\$A0,\$C8
7F74	5078A0C8F0	DB	\$50,\$78,\$A0,\$C8,\$F0,\$18,\$40,\$68,\$90,\$B8,\$E0,\$08,\$30,\$58,\$80,\$A8
7F84	D0F8204870	DB	\$D0,\$F8,\$20,\$48,\$70,\$98,\$C0,\$E8,\$10,\$38,\$60,\$88,\$B0,\$D8,\$00,\$28
7F94	5078A0C8F0	DB	\$50,\$78,\$A0,\$C8,\$F0,\$18,\$40,\$68,\$90,\$B8,\$E0,\$08,\$30,\$58,\$80,\$A8
7FA4	D0	DB	\$D0
7FA5	Q050A0	DB	LOW REL SCR,LOW (REL SCR+SCRLEN),LOW (REL SCR+(SCRLEN*2))
7FA8	F04090	DB	LOW (REL SCR+(SCRLEN*3)),LOW (REL SCR+(SCRLEN*4)),LOW (REL SCR+(SCRLEN*5))
7FAB	101010	DB	HIGH REL SCR,HIGH (REL SCR+SCRLEN),HIGH (REL SCR+(SCRLEN*2))
7FAE	101111	DB	HIGH (REL SCR+(SCRLEN*3)),HIGH (REL SCR+(SCRLEN*4)),HIGH (REL SCR+(SCRLEN*5))
7FB1	P1H		
7FB1	2021212121	DB	\$20,\$21,\$21,\$21,\$21,\$21,\$21,\$22,\$22,\$22,\$22,\$22,\$22,\$22,\$23,\$23
7FC1	2323232324	DB	\$23,\$23,\$23,\$23,\$24,\$24,\$24,\$24,\$24,\$24,\$25,\$25,\$25,\$25,\$25,\$25
7FD1	2526262626	DB	\$25,\$26,\$26,\$26,\$26,\$26,\$26,\$27,\$27,\$27,\$27,\$27,\$27,\$27,\$28,\$28
7FE1	2828282829	DB	\$28,\$28,\$28,\$28,\$29,\$29,\$29,\$29,\$29,\$29,\$2A,\$2A,\$2A,\$2A,\$2A,\$2A
7FF1	2A2B2B2B2B	DB	\$2A,\$2B,\$2B,\$2B,\$2B,\$2B,\$2B,\$2C,\$2C,\$2C,\$2C,\$2C,\$2C,\$2C,\$2D,\$2D
8001	2D2D2D2D2E	DB	\$2D,\$2D,\$2D,\$2D,\$2E,\$2E,\$2E,\$2E,\$2E,\$2E,\$2F,\$2F,\$2F,\$30,\$30
8010	3030303030	DB	\$30,\$30,\$30,\$30,\$30,\$30,\$31,\$31,\$31,\$31,\$31,\$31,\$32,\$32,\$32,\$32
8020	3232333333	DB	\$32,\$32,\$33,\$33,\$33,\$33,\$33,\$33,\$33,\$34,\$34,\$34,\$34,\$34,\$35,\$35
8030	3535353535	DB	\$35,\$35,\$35,\$35,\$35,\$36,\$36,\$36,\$36,\$36,\$36,\$37,\$37,\$37,\$37,\$37
8040	37	DB	\$37

'DLIST =8041'

8041	DLIST		
8041	707070	DB	\$70,\$70,\$70 ; 3x8 BLANK LINES
8044	462820	DB	\$46,\$28,\$20 ; ANTIC #4+LMS (SM=\$2000)
8047	0646642004	DB	\$06,\$46,\$64,\$20,\$04,\$04,\$84,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E
8057	8E0E8E0E8E	DB	\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E
8067	8E0E8E0E8E	DB	\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E
8077	8E0E8E0E8E	DB	\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E
8087	8E0E8E0E8E	DB	\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E
8097	8E0E8E0E8E	DB	\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E,\$8E,\$0E

DATA TABLES

```

80A7 8E0E8E0ECE DB $8E,$0E,$8E,$0E,$CE,$00,$30,$0E,$8E
80B0 0E8E0E8E0E DB $0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E
80B8 8E0E8E0E8E DB $8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E
80C8 8E0E8E0E8E DB $8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E,$8E,$0E
80DB 8E0E8E0E8E DB $8E,$0E,$8E,$0E,$8E,$0E,$90,$90,$90,$90,$90,$90,$90,$41
80E9 4180 DW DLIST

```

'AND ENDS AT 80EB'

```

      = 80EB      REFLOC EQU *
80EB = 00B1#      LOC     DLIST
      = 00B1      LOCSTR EQU *
00B1# 85E7        DLISTN STA LOW SVA
00B3# 86E9        STX     LOW SVX
00B5# A200        LDX     #$00
      = 00B6      SCRLPTR EQU *-1
00B7# B0010       LDA     RELSCR,X
      = 00B8      SCRLADR EQU *-2
00BA# 8D0AD4      STA     WSYNC
00BD# 8D1AC0      STA     $C01A
00C0# E8          INX
00C1# E050        CPX     #$80
00C3# D01F ^00E4 BNE     OK
00C5# C67D        DEC     SPEEDC
00E7# D019 ^00E2 BNE     OK2
00C9# A202        LDX     #$2
      = 00CA      SPEED EQU *-1
00CB# 867D        STX     SPEEDC
00CD# A200        LDX     #$00
      = 00CE      SCRLERM EQU *-1
00CF# B0A57F      LDA     SCRTBL,X
00D2# 85B9        STA     SCRLADR
00D4# B0AB7F      LDA     SCRTBH,X
00D7# 85B9        STA     SCRLADR+1
00D9# E8          INX
00DA# E006        CPX     #$06
00DC# D002 ^00E0 RNF     OK1
00DE# A200        LDX     #$00
00E0# 86CE        OK1    STX     SCRLERM
00E2# A200        OK2    LDX     #$00
00E4# 86B6        OK     STX     SCRLPTR
00E6# A900        LDA     #$00
      = 00E7      SVA     EQU *-1
00E8# A200        LDX     #$00
      = 00E9      SVX     EQU *-1
00EA# 40          BIL
      = 00EB      LOCEND EQU *
      = 8125      NEWORG EQU BEFLOC+(LOCEND-LOCSTR)
      MSG        'NEWORG=',NEWORG
      'NEWORG=8125'
00EB# = 8125      ORG     NEWORG
      LIST       -1

```

42 ERRORS, 680 Labels, 39A3h bytes not used. Program LWA = 9A83h.  
Last error occurred on Page 84

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 88  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

A1	67C8	30/53	30/58	31# 2				
AD1625	73AA	69#49						
ADD810	73B5	69/54	69#56					
ADDR	006B	1#16	30/13	30/18	30/23	31/ 2	32/29	32/34
		32/39	33/19	51/41	51/47	53/26	53/32	53/34
		53/37	67/24	67/26	67/38	67/40	67/53	67/55
		68/ 7	68/ 9					
ADDRM	006D	1#17	30/15	30/20	32/31	32/36	51/58	52/ 4
		67/28	67/30	67/42	67/44	67/57	67/59	68/11
		68/13						
ADDSCR	7389	18/19	20/40	25/42	63/38	69#30		
ARTS	6E13	56#30						
AUDC1	E801	3#17	16/26	16/30	17/ 4	21/25	23/49	25/ 8
		37/48						
AUDC2	E803	3#19	21/27	23/50	87/49			
AUDC3	E805	3#21	23/51	25/ 9	37/50			
AUDC4	E807	3#23	23/52	37/51				
AUDCTL	E808	3#14						
AUDE1	E800	3#16	16/17	17/ 7	21/21	21/42	25/ 2	25/30
AUDE2	E802	3#18	21/45					
AUDE3	E804	3#20	25/27					
AUDE4	E806	3#22						
BDEADC	006A	1#15	17/54	23/ 5	23/20			
BDIN10	6D40	48#36	48/46					
BDIN20	6D43	48#39	48/42					
BDINIT	6D25	37/32	48#23					
BEFLOC	80EB	36/ 2	87#11	87/50				
BEXPL	74DD	73/ 7	73#22					
BFIRE	7536	73/ 6	73#32					
BFIRWN	7544	73/12	73#34					
BIT	006E	1#18	30/11	32/27				
BLNKIT	634D	15/34	15#37					
BONUS	74C6	73/15	73#17					
BOOML	6659	25#10	25/12	25/32				
BOOZ	665B	25#11	25/14					
BORING	65F4	23#53	23/56					
BOUNCE	96BB	87#56						
BR1	72BA	65/61	68#30					
BSEAS	7232	23/ 8	43/16	67#23				
BSECNT	0081	1#36	45/11	45/13	45/17	45/42		
BSEI10	6CA9	45/10	45#15					
BSEI20	6CAE	45/12	45#22					
BSEI30	6CDE	45/26	45#45					
BSFIRE	6C9C	4/22	45# 7	68/19				
BSOUND	639B	4/38	17# 2					
BSPAIN	726E	44/58	67#52					
BSW100	6C96	44/11	44/32	44/40	44/42	44/50	44#56	
BSW020	6C35	43#36	43/50	43/61				
BSW030	6C3A	43/19	43#42					
BSW040	6C4A	43/21	43#54					
BSW050	6C57	43/25	43/38	43/49	44# 5			
BSW060	6C6E	44/ 8	44#24					
BSW070	6C78	44/20	44#30					
BSW080	6C7D	44/29	44#36					
BSW090	6C8A	44/19	44#46					
BSWORK	6C23	4/25	43#15					
BUCK	969D	68/24	68/25	87#56				
BUCKL	96A7	68/24	68/25	87#56				
BUCKR	96B1	68/24	68/25	87#56				
BUCKX	009A	1#57	22/14	22/54	37/41	43/37	43/45	43/57

1	BUCKY 009B	45/27	63/23	67/31	67/45	67/60	68/14	
2		1#58	22/16	22/56	37/39	44/15	44/25	44/31
3	BUKSH 72R1	45/31	67/47	68/16	72/32			
4		67/39	68/ 8	68#25				
5	BUKSL 72AE	67/37	68/ 6	68#24				
6	RYERYE 635D	15/36	15/38	15/42	15#44			
7	BYTE 0070	1#19	30/ 9	30/29	30/31	30/52	30/56	30/60
8		31/20	32/25	32/45	32/47	33/ 8	33/12	33/16
9		33/40						
10	CALI 0083	73#58	74/50	74/52	74/54	75/ 7	75/10	75/13
11		75/17	75/21	75/23	75/26	75/28	75/32	75/34
12		75/44	75/47	75/50	75/54	75/58	75/60	76/ 3
13		76/ 5	76/ 9	76/11	76/21	76/23	76/28	76/30
14		76/35	76/37	76/42	76/44	77/ 8	77/10	77/12
15		77/14	77/18	77/20	77/22	77/24	77/28	80/28
16		80/35	80/42	80/49	80/59	81/ 6	81/13	
17	CHACTL D401	3# 9	71/19					
18	CHART 0904	2#46	35/41	71/18				
19	CHBAS 0903	2#45	23/42	35/51	65/43	71/20		
20	CHBASE D409	3#10	71/21					
21	CHKH10 7367	68#60	69/ 7					
22	CHKH20 7376	69/ 4	69# 8					
23	CHKH30 7378	69#10	69/14					
24	CHKH40 7381	69/ 3	69#15					
25	CHKHSC 7365	35/48	68#58					
26	CHKITA 722B	67/10	67#14					
27	CHKVAL 6519	21#52	21/54					
28	CINIT 68E5	4/ 3	4/16	23/57	35# 7			
29	CKCR10 66CC	26/36	26#42					
30	CKCR20 66D0	26/41	26#45					
31	CKCR30 66E3	26/49	26#55					
32	CKCR40 66E7	26/54	26#58					
33	CKCR50 66EA	26/40	26/44	26/53	26/57	26#61		
34	CKCRSH 66AD	20/21	20/29	22/44	26#26			
35	CLERHI 6015	4# 7	4/11					
36	CLEV1 6DE6	49/55	49/56	49#57				
37	CLEV2 6DE8	49/55	49/56	49#58				
38	CLEV3 6DEA	49/55	49/56	49#59				
39	CLEV4 6DEC	49/55	49/56	49#60				
40	CLEV5 6DEE	49/55	49/56	49#61				
41	CL OOP 6956	36# 2	36/ 5					
42	n COL110 6487	20#37						
43	COL114 64C1	20/55	21# 5					
44	COL115 64CC	21/ 4	21#14					
45	COL120 6530	19/36	22#13					
46	COL130 6542	22#23	22/32					
47	COL140 6547	22#26	22/37	22/39	22/41	22/43	22/45	
48	n COL145 6550	22#33						
49	COL150 6551	22/25	22#35					
50	COL160 6595	23# 9	23/21					
51	COL170 65A2	23#15	23/17	23/19				
52	COL180 65C1	23/27	23#32					
53	COL185 65C4	23/23	23#34					
54	COL190 6600	20/23	24# 8					
55	COL195 6618	24/20	24#24					
56	COL200 661C	24/14	24#31					
57	n COL210 6628	24#37						
	COL220 6643	24/26	24#52					
	n COL260 6693	25#37						
	COL280 6695	25#39	25/44					



COL 220 6443 24/26 24#52  
n COL260 6693 25#37  
COL280 6695 25#39 25/44

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 90  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

COLNEG 0907	2#49	3/57	3/60	4/13	4/15			
COLDST 6011	3/59	4/2	4#4					
COLI10 6428	19#25	19/35						
COLI20 642F	19#29	20/19						
n COLI25 643B	19#37							
COLI30 643C	19/28	19#42						
COLI40 645D	20#2	20/16						
COLI50 646D	20#10	20/22	20/80					
COLI60 6475	20#17	21/9	24/48					
COLI65 647A	20/9	20#20						
COLI70 6482	20/5	20/7	20#28					
COLISN 6426	4/29	19#23						
COLOR0 0911	2#52	21/30	23/54	25/22	48/25	66/50	70/55	
COLOR1 0912	2#53	21/32	23/45	25/16	48/27	66/52	70/57	
COLOR2 0913	2#54	21/34	23/46	25/18	48/29	66/56	70/59	
COLOR3 0914	2#55	15/44	21/36	23/47	25/20	48/30	66/54	
	67/7	70/61						
COLOR4 0915	2#56	21/38	23/48	25/25	48/31	66/9	71/3	
COLRSH 6DE1	49/26	49#56						
COLRSL 6DDC	49/24	49#55						
CONFOT 7029	61/8	61/10	61#14					
CONSOL C01F	3#15	71/6						
CONVOK 607D	4/47	4/48	4#52					
CONVSP 606A	4/19	4#41	23/3	37/45	44/57			
CRASH 750C	73/5	73#26						
CREA10 6CE6	45/25	46#16						
CREA20 6CF3	46/15	46#23						
CREA30 6CE6	46/17	46#26						
CREA40 6CEF	46#32	46/36						
CREA5 6CE2	46#13	46/22						
CREATE 6CE0	6/18	7/27	8/15	9/36	10/52	12/16	22/52	
	28/11	28/15	42/39	46#11				
OSCORE 090D	2#51	47/9	18/24	18/35	36/24	36/25	36/26	
	36/27	65/10	65/19	66/31	66/40	68/61	69/11	
	69/33	69/34	69/36	69/37	69/39	69/40	69/42	
	69/43							
CVLOOP 6071	4#46	4/50						
DCODEH 79DA	51/36	53/21	84#8					
DCODEL 79D1	51/34	53/19	84#7					
DEADBK 6567	22#49	22/53	72/37					
DECIT 64BD	20/52	21#3						
DLILOC 00B1	2#30	35/32	35/34	36/3	87/12			
n DLI RTN 00B1	87#14							
DLIST 8041	35/54	35/56	86#52	87/6				
n DLISTH D403	3#12							
n DLISTL D402	3#11							
DMACTL D400	3#13	71/12						
U DOT 79E3	84/10	84/10	84/11	84/11	84/13	84/13	84/14	
	84/14	84/16	84/16	84/17	84/17	84/19	84/20	
	84/22	84/23	84/25	84/26				
DRAW 67EC	31#26	31/31						
DRIVER 602B	4#17	4/39	38/8					
ERASE 6818	32#17	55/41	67/35	67/49				
EXTILC 67B9	30/32	30/33	30#54					
n EXITRB 722A	67#13							
FIRDLY 00AF	2#28	36/11	45/41					
FIRFDG 00B0	2#29	19/48	19/54	36/14				
FRGTIT 6897	33/26	33#28	33/29					
FUEL 62C9	4/32	14#7						
n FUEL03 62C9	14#11							



1	FUEL05 62D1	14#16	14/34					
2	FUEL07 62D2	14/15	14#18					
3	FUEL10 6801	14/38	14#48	38/4				
4	FUEL ST 756C	73/11	73#40					
5	FUISOK 630C	14/55	14#57					
6	EULAMT 009C	1#59	14/12	14/13	14/14	14/27	14/30	14/31
7		14/33	14/35	14/37	14/40	14/41	14/42	14/50
8		15/ 7	15/26	23/24	23/25	23/26	37/19	37/21
9		37/22	71/47	71/48	71/49			
10	EJLP 6312	14#41	15/ 3					
11	FUXD 6335	15/15	15#26					
12	FJXX 6328	15/ 9	15#14	15/19				
13	GETNXT 73B6	61/30	61/37	61/44	70# 2			
14	GL1 67A5	30/41	30#43					
15	GL2 67B3	30/49	30#51					
16	GL3 6802	31/36	31#38					
17	GL4 6807	31/42	31#44					
18	GL5 6812	31/48	31#50					
19	GLDPO 6647	25# 2	25/ 4					
20	GLZ10 6DC6	49/46	49#48					
21	GLZ9 6DC9	49/43	49#45					
22	GRACTL C01D	3#24	85/31					
23	HERE 67D4	31#14	31/52					
24	HFIRE 7552	73/ 8	73#36					
25	HFRONT 0090	1#45	9/ 9	9/11	9/55			
26	HFRDLY 00A9	2#15	9/54					
27	HIGHT 0071	1#20	31/ 4	31/51	33/20	34/10		
28	HISCOR 0909	2#50	4/ 8	66/11	66/20	69/ 2	69/12	
29	U HOP1 7A0B	84/16	84/16	84/17	84/17			
30	U HOP2 7A0B	84/16	84/16	84/17	84/17			
31	U HOP3 7A0B	84/16	84/16	84/17	84/17			
32	U HOP4 7A0B	84/16	84/17					
33	U HOP5 7A0B	84/16	84/17					
34	HOPCNT 0084	1#39	12/ 9	12/11	12/33			
35	HOPDLY 00A3	2# 9	12/32					
36	HOPF10 61A4	9/ 8	9#12	9/37				
37	HOPF20 61A5	9/10	9#14					
38	HOPF30 61A7	9#16	9/26					
39	HOPF35 61AE	9#20	9/31	9/33				
40	HOPF40 61BA	9/19	9#28					
41	HOPF50 61F4	9/27	9#53					
42	HOPFIR 619A	4/33	9# 6					
43	HOPP10 628A	12/ 8	12#12	12/17				
44	HOPP20 628B	12/10	12#14					
45	n HOPP30 6292	12#18						
46	n HOPP40 629D	12#24						
47	n HOPP50 62A8	12#29						
48	HOPPER 627E	4/28	12# 5					
49	HOPSPD 00A4	2#10	12/30					
50	HSCOUT 71AE	66#11	66/28					
51	n INC01 749E	72#46						
52	INC01Q 74A4	72/48	72#50					
53	INC21Q 74AD	72/55	72#57					
54	n INC23 74A7	72#53						
55	INDR10 73C1	70/10	70#12					
56	INDRPT 73BB	60/57	62/10	62/11	62/18	70/ 3	70# 8	
57	n INIT 6904	35#26						
	INS010 7388	69/23	69#25					
	INSDPT 7382	16/12	16/24	69#21				
	INVDAT 8125	87#56	87/56	87/56	87/56	87/56	87/56	

INSDPT 7382 16/12 16/24 69#21  
INVDAT 8125 87#56 87/56 87/56 87/56 87/56 87/56 87/56

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 92  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

	87/56	87/56	87/56	87/56	87/56	87/56	87/56
	87/56	87/56	87/56	87/56	87/56	87/56	87/56
	87/56	87/56	87/56	87/56	87/56	87/56	87/56
	87/56	87/56	87/56	87/56	87/56	87/56	87/56
	87/56	87/56	87/56	87/56	87/56	87/56	87/56
	87/56	87/56	87/56	87/56	87/56	87/56	87/56
JAMIT 62C8	13/ 9	13#18					
JOY100 749B	72/34	72#36					
JOYS10 7443	71/50	71#52					
JOYS90 7490	71/51	72#29					
JOYSTK 7434	4/18	71#44					
JS1 7453	71/55	71#60					
JS2 746D	72/ 7	72#12					
JSDATA 0060	1#12	40/13	43/18	44/ 6	44/ 9	45/ 8	71/46
	71/56	71/58	72/ 2	72/ 4	72/ 8	72/10	72/14
	72/16	72/20	72/22	72/25	72/27	72/31	
JSJAM 7477	72/11	72/13	72#17				
JSJAM1 7484	72/19	72#23					
JSJAM2 748F	72/24	72#28					
JUMP 0080	73#55	77/30	77/35	77/38	79/49	79/52	79/55
	79/58	79/61	80/ 4	80/ 7	80/10	80/13	80/16
	80/19	80/22	80/30	80/37	80/44	80/51	80/56
	80/61	81/ 8	81/15	82/60	83/ 8		
KBCODE E809	3#36	66/57	67/ 8				
KBELG 007A	1#26						
KILL 0082	73#57	74/56	75/37	76/14	77/ 3	79/22	79/44
	82/19	82/57					
LASTBS 0080	1#35	43/24	43/44	43/56	67/36	68/ 5	
LEVELD 6997	36/44						
LOADPC 0081	73#56	76/48	80/28	80/35	80/42	80/49	80/56
	80/59	81/ 6	81/13				
LOGEND 00EB	87#48	87/50					
LOCSTR 00B1	87#18	87/50					
LVLDT 009F	2# 4	2/19	36/60				
LVLFLG 007B	1#27	36/20	55/28	55/31	63/44		
LVLQVR 64CC	21#13	55/32					
MASK 0072	1#21	30/16	30/21	30/47	30/48	30/50	31/27
	31/34	31/35	31/37	32/32	32/37	33/ 3	33/ 4
	33/ 6	33/47	33/53	33/54	33/56		
MAXLVL 0014	36/34	39#58					
U MBUCK 72B4	68/27	68/28					
U MBUCKL 72B4	68/27	68/28					
U MBUCKR 72B4	68/27	68/28					
MBUKSH 72B7	67/43	68/12	68#28				
MBUKSL 72B4	67/41	68/10	68#27				
MCODEH 7A8C	51/53	84#36					
MCODEI 7A83	51/51	84#35					
U MDQT 7A95	84/38	84/38	84/39	84/39	84/41	84/41	84/42
	84/42	84/44	84/44	84/45	84/45	84/47	84/47
	84/47	84/47	84/47	84/47	84/47	84/47	84/48
	84/48	84/48	84/48	84/48	84/48	84/48	84/48
	84/50	84/51	84/53	84/54			
MDROID 7ABD	84/35	84/36	84#44				
MEANEG 007C	1#28	36/23	36/39	36/42	37/ 9		
MESHT 7AD1	84/35	84/36	84#47				
MFRCNT 0092	1#47	8/ 3	8/ 5	8/10			
MFRDLY 00AB	2#17	8/ 9					
U MHOP1 7ABD	84/44	84/44	84/45	84/45			
U MHOP2 7ABD	84/44	84/44	84/45	84/45			
U MHOP3 7ABD	84/44	84/44	84/45	84/45			

1	U MHOP4 7ABD	84/44	84/45					
2	U MHOP5 7ABD	84/44	84/45					
3	MTNSPD 00A7	2#13	37/7	37/10	37/43	44/49	44/51	
4	MTI100 615B	8/4	8#7					
5	MOTH10 6102	7#9	7/28					
6	MOTH20 6103	7/8	7#11					
7	MOTH30 6105	7#13	7/22					
8	MOTH40 612C	7/33	7#36					
9	MOTH50 612E	7/35	7#38	7/52				
10	MOTH60 6141	7/31	7#47					
11	MOTH70 614C	7/16	7#56					
12	MOTH80 6153	7#60	8/13	8/16				
13	MOTH90 6154	7/52	8#2					
14	MOTHER 60FC	4/35	7#5					
15	MOUDAT 970B	35/50	87#56					
16	MOV100 6FC9	57/28	57/40	57/42	58/21	58/25	58#29	
17	MOVE 6E7F	4/23	22/50	23/10	54#47	55/51	55/56	
18	MOVE20 6E8B	54#54	55/13					
19	MOVE30 6EA7	54/56	54/58	55/3	55#7			
20	MOVE35 6EB7	55#18	55/27					
21	MOVE40 6ED3	55/15	55#34					
22	MOVE50 6EE5	55/40	55#42					
23	MOVE60 6EF6	55/48	55#50					
24	MOVE70 6EF9	55/45	55#52					
25	MOVPO3 6F09	56#25	56/29					
26	MOVPO5 6F14	56/27	56#34					
27	MOVPO6 6F22	56/37	56#41					
28	MOVPO7 6F31	56/40	56#49					
29	MOVPO10 6F51	57#10	57/44					
30	MOVPO20 6F72	57/12	57#35					
31	MOVPO40 6F88	57/16	57/29	57#49				
32	MOVPO50 6FA5	57/60	58#4	58/7				
33	MOVPO60 6FAA	58#8						
34	MOVPO70 6FB2	58/10	58#14					
35	MOVPO80 6FC2	58/19	58#23					
36	MOVPO90 6FC5	57/52	57/55	58/22	58#26			
37	MOVPO5 6F03	55/44	56#19					
38	U MPOL1 7A95	84/38	84/39					
39	U MPOL2 7A95	84/38	84/39					
40	U MPOL3 7A95	84/38	84/39					
41	U MPOL4 7A95	84/38	84/39					
42	U MPOL5 7A95	84/38	84/38	84/39	84/39			
43	U MPOL6 7A95	84/38	84/38	84/39	84/39			
44	MPOLE 7A95	84/35	84/36	84#38				
45	MPSH1 7B0D	84/35	84/36	84#56				
46	U MSAU1 7AA9	84/41	84/41	84/42	84/42			
47	U MSAU2 7AA9	84/41	84/41	84/42	84/42			
48	U MSAU3 7AA9	84/41	84/41	84/42	84/42			
49	U MSAU4 7AA9	84/41	84/41	84/42	84/42			
50	MSAUC 7AA9	84/35	84/36	84#41				
51	U MSHAD0 723A	67/27	67/29	67/56	67/58			
52	MSHOT 7AE5	84/35	84/35	84/36	84/36	84#50		
53	U MSHQT1 7AE5	84/50	84/51					
54	U MSHOT2 7AD1	84/47	84/48	84/50	84/51			
55	U MSHOT3 7AD1	84/47	84/48	84/50	84/51	84/56	84/56	84/56
56		84/56	84/56	84/56	84/56	84/56	84/56	84/56
57		84/57	84/57	84/57	84/57	84/57	84/57	84/57
	U MSHOT4 7AE5	84/50	84/51					
	U MSHOT5 7AE5	84/50	84/51					

U MSHOT4 7AE5  
U MSHOT5 7AE5

84/37 84/57 84/57 84/57  
84/50 84/51 84/50 84/51

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 94  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

U MSHOT6 7AE5 84/50 84/50 84/50 84/50 84/51 84/51 84/51

MSTAR 7B21 84/51 84/51 84/51 84/51 84/51 84/51 84/51  
MSTAR 7B21 84/59 84/59 84/59 84/59 84/59 84/59 84/59

84/59 84/59 84/59 84/60 84/60 84/60 84/60  
84/60 84/60 84/60 84/60 84/60 84/60 84/60

MTN130 6BCC 41/ 2 41/27 41#40  
MTN0NT 0098 1#55 37/31 40/60 40/61 41/ 4  
MTNSPD 0096 1#53 22/61 37/47 40/18 40/33 40/40 40/46  
40/52 41/ 5 57/58

MTNW05 6B44 40/ 9 40#11  
MTNW10 6B5C 40/19 40#25  
MTNW30 6B66 40/14 40#32  
MTNW40 6B72 40/16 40#39  
MTNW50 6B7B 40/22 40/24 40/28 40/30 40/36 40/38 40/43  
40#45

MTNW60 6B7D 40#50  
MTNW70 6B87 40/53 40#57  
MTNW80 6B86 41/ 6 41#14  
MTNW90 6B82 41/10 41/18 41/18 41#24 48/60  
MTNW40 6B8C 41#30 41/39  
MTNWRK 6B3D 4/20 40# 6

MTOK 6BC7 41/35 41#37  
MTRBAK 0087 74# 2 78/19 78/60  
MTRHTB 6A44 8/23 24/15 25#50  
MTRSPD 00A5 2#11 7/44  
MTSTRT 0097 1#54 37/29 41/ 7 41/ 8 41/12 41/15 41/16  
41/20 41/29

MULT 73C2 14/24 36/47 56/46 70#30  
MULT10 73D0 70#32 70/46  
MULT20 73D4 70/40 70#42  
MULT30 73DD 70/32 70#49  
MVXY 6FD4 57/27 57/39 59# 4  
MVXY10 6EE0 59#11 59/17 59/25 59/33

MVXY20 6EE2 59/ 6 59#13  
MVXY30 6EE3 59/ 5 59/ 8 59#15  
MVXY40 6EE8 59/ 9 59#26 59/34

MVXY50 6FFB 59/20 59#29  
U MZOR1 7AF9 84/53 84/54  
U MZOR2 7AF9 84/53 84/53 84/54 84/54  
U MZOR3 7AF9 84/53 84/53 84/54 84/54  
U MZOR4 7AF9 84/53 84/53 84/54 84/54  
U MZOR5 7AF9 84/53 84/53 84/54 84/54

MZORBA 7AF9 84/35 84/36 84#53  
NEWL05 6997 36/35 36#43  
NEWL10 69B4 36#58 37/ 3  
NEWL20 69D5 37/17 37#23  
NEWLVI 69B5 22/ 5 23/31 25/46 36#32  
NEWORG 8125 87#50 87/52 87/54

NEWP10 6A11 37#54 37/58  
NEWPLY 69D5 23/33 37#27  
NM1EN D40E 3#25 23/36 35/11 35/53 49/ 3 49/13 49/17  
65/45

NOGAIN 6ED2 55/29 55#33  
NXT100 70C3 62/52 62#54 63/ 4 63/ 7

NXT110 70C6 62/45 62#56  
NXT120 70C9 62/48 62#59  
NXT130 70D3 63/ 3 63# 5

NXT140 70D8 61/21 63#15  
NXT150 70E6 63/17 63#22

NXT160 70E7	63/27	63#31					
NXT170 70FB	63/29	63#34					
NXT175 710E	63/42	63#45					
NXT180 711E	63/30	63/33	63#57				
NXT190 7125	61/22	64# 5					
NXT200 712E	64/ 7	64#10					
NXT210 7131	64/ 9	64#12					
NXTR10 700B	60#58	61/32	61/39	61/50			
NXTR20 7012	60/60	61# 2	61/ 5				
NXTR21 7039	61/17	61#23					
NXTR25 703C	61/20	61#29					
NXTR30 7045	61/16	61#36					
NXTR40 704E	61/15	61#43	61/60				
NXTR50 705E	61/18	61#55					
NXTR60 706B	61/19	62# 5					
NXTR70 707E	60/61	62#16					
NXTR80 70A2	62/25	62#33					
NXTR90 70AD	62/22	62/30	62/32	62/37	62#42		
NXTRPT 700B	56/28	57/41	60#56	62/12	63/21	63/60	64/17
OBJTBL 0600	3#41	6/25	6/42	6/44	7/14	7/44	7/43
	7/45	7/57	8/11	8/20	8/21	8/26	8/27
	8/28	8/29	8/31	9/17	9/29	9/41	9/44
	9/45	9/46	9/47	9/48	9/50	10/16	10/32
	10/41	10/57	10/58	10/60	11/ 3	11/ 5	11/15
	12/26	12/28	12/31	19/26	19/44	19/46	20/ 3
	20/38	20/42	20/47	22/24	22/36	22/55	22/57
	24/12	24/17	24/34	24/36	24/43	24/45	26/28
	26/33	26/46	28/19	28/36	28/43	28/45	28/47
	28/49	28/51	28/53	37/56	42/35	42/42	42/45
	42/48	45/30	45/34	45/36	45/38	46/16	46/28
	46/33	51/22	51/29	51/32	51/38	51/49	51/55
	52/ 6	52/ 8	53/17	53/23	54/55	54/57	55/ 2
	55/ 4	55/20	55/21	55/36	55/38	55/55	56/26
	56/35	56/44	56/51	56/60	56/61	57/18	57/26
	57/37	57/50	57/53	58/17	58/18	59/21	59/24
	59/30	59/32	61/31	61/38	62/17	62/26	62/31
	62/34	62/38	62/49	62/55	62/60	63/18	63/26
	64/14	64/16					
OK 00E4	87/24	87#42					
OK1 00E0	87/38	87#40					
OK2 00E2	87/26	87#41					
OKP1 67E9	31/22	31#24					
OPTLOP 7199	65#61	66/ 7					
PIH 7FB1	31/16	33/36	86#39				
PIL 7F15	31/18	33/38	86#24				
PACTI 0302	3#26	35/43					
PAINT 675A	29# 3	55/49	68/ 4	68/18			
PATH0 7616	74/ 7	74#49					
PATH1 7629	74/ 8	75# 2					
PATH10 7604	74/17	77# 7					
PATH11 76F6	74/18	77#17					
PATH12 7718	74/19	77#27	77/31	80/57			
PATH13 7725	74/20	77#34	77/36				
PATH14 772A	74/21	77#37	77/39				
PATH15 772E	74/22	77#40					
PATH16 7768	74/23	78#21					
PATH17 77A1	74/24	79# 2					
PATH18 77B8	74/25	79#24					
PATH19 77CF	74/26	79#48	79/50				
PATH2 7661	74/ 9	75#39					



PATH19 77CF 74/26 79#48 79/50  
PATH2 7661 74/ 9 75#39

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 96  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

PATH20	77D4	74/27	79#51	79/53				
PATH21	77DB	74/28	79#54	79/56				
PATH22	77E2	74/29	79#57	79/59				
PATH23	77E7	74/30	79#60	80/ 2				
PATH24	77EE	74/31	80# 3	80/ 5				
PATH25	77E5	74/32	80# 6	80/ 8				
PATH26	77FA	74/33	80# 9	80/11				
PATH27	7801	74/34	80#12	80/14				
PATH28	7808	74/35	80#15	80/17				
PATH29	780D	74/36	80#18	80/20				
PATH3	7692	74/10	76#19					
PATH30	7814	74/37	80#21	80/23				
PATH31	781B	74/38	80#27	80/31				
PATH32	782A	74/39	80#34	80/38				
PATH33	7839	74/40	80#41	80/45				
PATH34	7848	74/41	80#48	80/52				
PATH35	7857	74/42	80#55					
PATH36	785C	74/43	80#58	81/ 2				
PATH37	786B	74/44	81# 5	81/ 9				
PATH38	787A	74/45	81#12	81/16				
PATH4	76A1	74/11	76#26					
PATH5	76A9	74/12	76#33					
PATH6	76B1	74/13	76#40					
PATH7	76B9	74/14	76#47					
PATH8	76D3	74/15	77# 2					
PATH9	76C1	74/16	76#52					
U PAUSE	6061	4/37						
PBCTL	D303	3#27	35/44					
PCSCAL	6E50	19/51	26/31	53#14				
PCXY05	6DE7	51#25	51/31					
PCXY10	6DEA	51/24	51#28					
PCXYAC	6DF0	51#21	55/39	55/47				
PDL0	0916	2#57	71/14	71/53				
PDL1	0917	2#58	71/16	72/ 5				
PFIRE	755E	73/ 9	73#38					
PFRONT	0091	1#46	10/ 8	10/10	10/28			
PERDLY	00AA	2#16	10/27					
PLYLVL	00AE	2#24	21/15	23/28	23/30	25/45	36/22	36/33
		36/38	36/45	37/15	49/20			
PLYTBL	6A25	36/50	36/54	38#30	39/58			
PMBASE	D407	3#28						
POL1	96CF	84/10	84/11	87#56				
POL110	626E	10/37	11#11					
POL2	96D9	84/10	84/11	87#56				
POL3	96E3	84/10	84/11	87#56				
POL4	96ED	84/10	84/11	87#56				
POL5	96F7	84/10	84/10	84/11	84/11	87#56		
POL6	9701	84/10	84/10	84/11	84/11	87#56		
POLCNT	0082	1#37	28/ 8	28/23	28/60			
POLDLY	00A0	2# 6	28/59					
POLE10	670E	28/ 7	28/12	28#20	28/24			
POLE20	670F	28/ 9	28#22					
POLE30	6714	28/16	28#28					
POLE35	6716	28#30						
POLE40	6716	28#31	28/34	28/40	28/42			
POLES	66ED	4/26	28# 4					
POLF10	6205	10/ 7	10#11					
POLF20	6206	10/ 9	10#13					
POLF30	6208	10#15	10/25					
POLF40	620F	10#19	10/34	10/40	11/10	11/14		

1	POLF45 6218	10#26	10/53					
2	POLF50 6210	10/18	10#30					
3	POLF60 6239	10#44	11/17					
4	POLF70 623E	10/43	10#47	11/18				
5	POLF80 6240	10/46	10#49					
6	POLF85 6256	10#59						
7	POLF90 6269	11# 8						
8	POLFIR 61F9	4/34	10# 4					
9	POLFND 008D	1#42	63/16	63/20	63/28	63/32	63/59	
10	POLPAS 0086	73#61	75/31	76/ 8				
11	PRECHN 74D1	17/29	73#20					
12	PJCNT 000E	2#19	36/46	37/ 2	39/58			
13	PTHPTR 0020	1#11	47/18	47/20	57/13	60/59	61/46	61/47
14		61/49	61/56	61/58	62/ 7	62/ 9	62/19	70/ 4
15		70/ 9	70/11					
16	PTHTR 75C8	47/17	47/19	74# 6				
17	PXLPAS 75B4	73/14	73#49					
18	Q4 7D15	30/ 8	32/24	85#46				
19	QAZK 6D5D	48#53	48/56					
20	QAZL 6D52	48#48	48/51					
21	R4 7E15	80/10	32/26	86# 3				
22	RANDOM 880A	3# 7	6/21	6/28	7/32	7/48	10/35	12/19
23		12/27	21/29	21/31	21/33	21/35	21/37	21/52
24		24/38	25/15	25/17	25/19	25/21	25/23	28/32
25		42/18	42/23	64/ 6				
26	R8OW 7209	66#59	67/12	67/16				
27	RCLIP 67BA	30/30	30#55					
28	RELSCR 1000	3#32	49/32	49/34	86/35	86/35	86/35	86/36
29		86/36	86/36	86/37	86/37	86/37	86/38	86/38
30		86/38	87/18					
31	RET 0084	73#59	74/58	77/33	80/33	80/40	80/47	80/54
32		81/ 4	81/11	81/18	81/22	81/25	81/28	81/30
33		81/33	81/36	81/39	81/41	82/ 4	82/41	83/ 5
34		83/18						
35	RFHG 6318	14/59	15# 4					
36	RICQCH 7528	73/10	73#30					
37	RPTTBL 6C1F	42/41	42#55					
38	RSCOUT 71D1	66#31	66/48					
39	RSPONS 009F	2# 5	40/ 7	40/35	40/42			
40	RTNADD 0800	3#31	61/57	61/59	62/ 6	62/ 8		
41	U SAU1 79F7	84/13	84/13	84/14	84/14			
42	U SAU2 79F7	84/13	84/13	84/14	84/14			
43	U SAU3 79F7	84/13	84/13	84/14	84/14			
44	U SAU4 79F7	84/13	84/13	84/14	84/14			
45	SAUC10 6BD9	42/ 9	42#18	42/31				
46	SAUC20 6BDA	42/11	42#15					
47	SAUC30 6BF2	42#28	42/50					
48	SAUC40 6C1A	42#51						
49	SAUCER 6BCD	4/27	42# 6					
50	SAUCNT 0083	1#38	42/10	42/12	42/53			
51	SAUDLY 00A1	2# 7	42/52					
52	SAUSPD 00A2	2# 8	42/47					
53	SAUX 6BDE	42#18						
54	SCDAT 6D76	48/48	49# 6					
55	SCONVE 6084	4/46	4#58					
56	SCOR01 6401	17/55	18#23					
57	SCOR02 6413	18/32	18#34					
	SCOR05 6422	18/30	18#41					
	SCORE 63DB	4/30	17#53	23/12				
	SCOREC 0061	1#13	17/56	18/ 8	18/ 9			

SCORE 63DB  
SCOREC 0061

4/30 17#53 23/12  
1#13 17/56 18/ 8 18/ 9

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 98  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

SCRATC 008F	1#44	15/31	15/33				
SCRONT 0095	1#52						
SCREEN 0074	1#22	31/17	31/21	31/23	31/26	31/29	31/37
	33/41	33/43	33/46	33/48			
SCREND 7D15	49/49	49/52	85#44				
SCRLAD 00B8	87#19	87/33	87/35				
SCRIEN 0050	85#18	86/35	86/35	86/36	86/36	86/36	86/37
	86/37	86/38	86/38	86/38			
SCRLFR 00CE	87#31	87/40					
SCRIPT 00B6	49/11	87#17	87/42				
SCROL0 7B35	49/28	49/30	85#13	85/18			
SCRQL1 7B85	85#19						
SCRQL2 7B05	85#24						
SCRQL3 7C25	85#29						
SCRQL4 7C75	85#34						
SCRQL5 7CC5	85#39						
SCRSET 0094	1#51						
SCRSPD 0099	1#56	4/44	14/19	17/ 5	18/ 4	23/ 2	37/44
	44/37	44/39	44/41	44/47	44/48	44/52	56/45
SCRIBH 7FAB	86#87	87/34					
SCRIBL 7FA5	86#35	87/32					
SCRUIO 7143	65#10	65/27					
SCRUPD 713F	18/42	65# 7					
SDLOP 6D8E	49#14	49/16					
SDLSTH 0901	2#43	23/38	35/57	65/41	71/ 9		
SDLSTI 0900	2#42	23/40	35/55	65/39	71/ 7		
SDMCTL 0902	2#44	65/47	71/11				
SDSTAT 0089	1#41	16/ 9	16/29	17/19	17/34	37/52	
SEICOL 6D99	37/33	49#20					
SETDLI 6D82	37/37	49# 9					
SETPH 6D0F	6/40	7/39	8/18	9/39	10/55	12/23	20/44
	20/49	22/59	24/42	28/55	28/58	42/33	45/40
	47# 7	64/13					
SETSPD 0085	73#60	76/49	76/53				
SHADOW 96C5	67/23	67/25	67/52	67/54	87#56		
SHAPE 0076	1#23	30/14	30/19	30/39	30/40	30/42	31/28
	31/46	31/47	31/49	32/30	32/35	32/55	32/56
	32/58	34/ 5	34/ 6	34/ 8			
SHIPS 62B2	13# 5	18/40	37/42				
SHOK 62BE	13/11	13#13					
SHOPD 62C2	13#15	13/17					
U SHOT1 7A1E	84/19	84/19	84/19	84/19	84/19	84/19	84/19
	84/20	84/20	84/20	84/20	84/20	84/20	84/20
	84/22	84/23					
U SHOT2 7A1F	84/19	84/20	84/22	84/23			
U SHOT3 7A1F	84/19	84/20	84/22	84/23	84/28	84/28	84/28
	84/28	84/28	84/28	84/28	84/28	84/28	84/28
	84/29	84/29	84/29	84/29	84/29	84/29	84/29
	84/29	84/29	84/29				
U SHOT4 7A33	84/22	84/23					
U SHOT5 7A33	84/22	84/23					
U SHOT6 7A33	84/22	84/22	84/22	84/22	84/23	84/23	84/23
	84/23						
SHPLFT 00AD	2#23	13/ 6	18/31	18/33	23/22	36/12	
SKCTL E80F	3#33	35/47					
SKIPMT 6D70	48/59	49# 2					
SKPDLI 69EC	37/36	37#38					
SKSTAT E80E	3#35	72/17					
U SLIST 65C9	23/37	23/39					
SNDAGE 0085	1#40	16/18	16/21	17/36			

1	SNDI10 63AB	17#18	17/25					
2	SNDI20 63B4	17/20	17#23					
3	SNDI50 63BF	17/22	17#31					
4	SNDINI 63AB	8/33	9/52	11/7	15/30	17#15	18/39	21/8
5		23/7	24/47	45/44	63/52			
6	SNDETR 0062	1#14	16/13	16/25	17/43	17/45	69/22	69/24
7	SNDTBI 74B0	17/42	17/44	73#4				
8	SOUN10 6365	16#8	16/35					
9	SOUN20 636A	16#11	16/27					
10	SOUN30 6384	16/16	16#23					
11	SOUN40 638F	16/14	16#28					
12	SOUN50 6395	16/10	16/22	16#31				
13	SOUND 6361	4/21	4/24	16#5	23/11			
14	SPEED 00CA	4/52	36/8	87#28				
15	SPEEDC 007D	1#29	36/7	87/25	87/29			
16	SPREAD 00A8	2#14	28/39					
17	SPRTA 0905	2#47						
18	SPRTB 0906	2#48						
19	SSCORE 008E	1#43	18/27	18/37				
20	STARS 60B9	4/36	6#6					
21	START 6000	3/46	3#53					
22	U STER 7A6F	84/31	84/31	84/31	84/31	84/31	84/31	84/31
23		84/31	84/31	84/31	84/32	84/32	84/32	84/32
24		84/32	84/32	84/32	84/32	84/32	84/32	
25	STR10 60BF	6#10	6/16	6/19				
26	STR20 60C0	6/9	6#12					
27	STR30 60CE	6#20	6/22					
28	STR40 60DA	6#27	6/30	6/32				
29	STR50 60EE	6/36	6#39					
30	STRCNT 0093	1#48	6/13	6/14				
31	SUB0 7622	74/51	74/53	74/55	74#57			
32	SUB1 79B1	75/8	75/11	75/14	75/22	75/27	75/38	82#59
33	SUB10 7978	76/24	76/45	82#5				
34	SUB11 7989	76/36	76/43	82#21				
35	SUB12 79A0	76/31	76/38	82#43				
36	SUB14 7889	77/15	81#20					
37	SUB15 78A8	77/23	81#23					
38	SUB16 78C7	77/11	81#26					
39	SUB17 78E6	77/19	81#29					
40	SUB19 78F5	77/25	81#31					
41	SUB20 7914	77/13	81#34					
42	SUB21 7933	77/21	81#37					
43	SUB22 7952	77/9	81#40					
44	SUB23 771E	77/29	77#32					
45	SUB24 7823	80/29	80#32					
46	SUB25 7832	80/36	80#39					
47	SUB26 7841	80/43	80#46					
48	SUB27 7850	80/50	80#53					
49	SUB28 7864	80/60	81#3					
50	SUB29 7873	81/7	81#10					
51	SUB3 79B8	75/18	75/29	75/35	83#2			
52	SUB30 7882	81/14	81#17					
53	SUB4 79BC	75/24	82/61	83#4				
54	SUB5 79C1	75/45	75/48	75/51	75/52	76/4	76/10	83#7
55	SUB7 79C8	75/55	76/6	76/12	83#10			
56	SUB8 79CC	75/61	83/9	83#12				
57	SUB9 7961	76/22	76/29	81#44				
	SVA 00E7	87/14	87#44					
	SVX 00E9	87/15	87#46					
	SYSTAT 00AC	2#18	6/7	7/6	7/29	9/7	10/5	10/38

SVA 00E7 87/14 87#44  
 SVX 00E9 87/15 87#46  
 SYSTAT 00AC 2#18 6/ 7 7/ 6 7/29 9/ 7 10/ 5 10/38

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 100  
 BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

		11/12	12/ 6	20/50	20/58	20/61	25/ 6	25/26
		25/31	28/ 5	36/21	37/34	41/25	42/ 7	48/57
		66/58	67/ 9					
	TDRDIB 7A0B	84/ 7	84/ 8	84#16				
	TESHT 7A1F	84/ 7	84/ 8	84#19				
	TGRN 635B	15/40	15#43					
	TH 67E7	31/15	31#32					
	TIDAT 6D7C	48/53	49# 7					
	TIMER 0078	1#24	23/44	23/53	66/61	67/ 6	71/22	
	TITLE 7163	35/49	65#35					
	TLIST 7346	65/38	65/40	68#49	68/53			
	TPOLE 79E3	84/ 7	84/ 8	84#10				
	TPSHT 7A5B	84/ 7	84/ 8	84#28				
	TSAUC 79F7	84/ 7	84/ 8	84#13				
	TSHOT 7A33	84/ 7	84/ 7	84/ 8	84/ 8	84#22		
	TSIAR 7A6E	84/ 7	84/ 8	84#31				
	TYEL 6352	15/28	15#39					
	TZORBA 7A47	84/ 7	84/ 8	84#25				
	UDSCOL 6D9E	21/55	49#24					
	UFOCNT 00A6	2#12	5/ 5	20/53	20/54	20/57	21/ 8	63/41
	UFADSP 608B	5# 4	21/ 6	38/ 5	63/50			
	UEQLX 60A1	5#15	5/19	5/21				
	UEOOK 609B	5#12	5/25					
	UFOXT 60B0	5/ 6	5/17	5#22				
	UEPT 60B1	5/ 8	5#23					
	VRIRTN 73DF	35/36	35/38	70#53				
	VDCUNT D40B	3#34	21/40	25/11	33/28	49/14	66/59	
	VDSLST 0206	3#29	35/33	35/35				
	VVBLKI 0202	3#30	35/37	35/39				
	WAITVB 6504	21#40	21/41	21/47				
	WARP 64CE	21#19						
	WARP1 64E4	21#29	22/ 4					
	WARPP5 64D2	21#21	21/23					
	WIDTH 007E	1#31	30/27	30/35	30/43	31/33	31/38	31/44
		32/43	32/51	32/59	33/52	33/57	34/ 3	
	WIDTHC 007E	1#30	30/34	30/37	30/45	30/55	30/61	31/24
		32/50	32/53	32/61	33/11	33/17	33/44	
	WSYNC D40A	3# 8	67/ 4	87/20				
	XLOOP 6DB8	49#36	49/50	49/53				
	XSHIFT 6ECB	57/56	58#32					
	YAXIS 745D	71/59	71/61	72# 5				
	YCODD 0079	1#25	30/ 7	30/26	32/23	32/42	33/21	
	ZA1 6886	33/ 9	33/14	33#18				
	ZAPIT 68ED	35#13	35/22					
	ZDRAW 68BB	33#46	33/50					
	ZEXITL 6877	32/48	32/49	33#10				
	ZGL1 6863	32/57	32#59					
	ZGL2 6871	33/ 5	33# 7					
	ZGL3 68CF	33/55	33#57					
	ZGL4 68D4	33/61	34# 8					
	ZGL5 68DF	34/ 7	34# 9					
	ZHERE 68A3	33#34	34/11					
	ZOD 6327	15#13						
	ZODS 631B	15# 7						
	ZOKP1 68B8	33/42	33#44					
	U ZOR1 7A47	84/25	84/26					
	U ZOR2 7A47	84/25	84/25	84/26	84/26			
	U ZOR3 7A47	84/25	84/25	84/26	84/26			
	U ZOR4 7A47	84/25	84/25	84/26	84/26			
	U ZOR5 7A47	84/25	84/25	84/26	84/26			



ZDRBD	7577	25/29	73/13	73#43
ZRCLIP	6878	32/46	33#11	
ZTH	6884	33/35	33#51	
ZXDBL	7185	65#50	65/59	
ZXCPL	718A	65#53	65/55	